Complexity of Clique-Guarded First-Order Logic with Counting

Steffen van Bergerem
Humboldt-Universität zu Berlin, Germany

Johannes Friedrich Lange
Humboldt-Universität zu Berlin, Germany

Nicole Schweikardt
Humboldt-Universität zu Berlin, Germany

Abstract

We introduce *clique-guarded first-order logic with counting* (cgFOC), a fragment of the first-order logic with counting FOC [Kuske and Schweikardt, LICS 2017], and we study the complexity of this fragment. In particular, we prove computable upper bounds on the Vapnik–Chervonenkis (VC) dimension of cgFOC formulas and on the graph dimension of cgFOC counting terms on nowhere dense classes of relational structures. Furthermore, we show algorithmic metatheorems for cgFOC for query answering, enumeration, and probably approximately correct (PAC) learning for Boolean and multiclass classification problems on classes of locally bounded expansion. On the other hand, we show that a slight extension of cgFOC is already intractable on trees.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Theory of computation → Complexity theory and logic; Computing methodologies → Supervised learning

Keywords and phrases First-order logic with counting, VC dimension, graph dimension, algorithmic metatheorems, enumeration, nowhere dense, locally bounded expansion, PAC learning

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.20

Related Version *Full Version*: `https://arxiv.org/abs/2606.24848v1` [6]

Funding This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 541000908 (gefördert durch die Deutsche Forschungsgemeinschaft (DFG) – Projektnummer 541000908).

1 Introduction

The complexity of first-order logic (FO) has been studied intensively for several complexity measures over the last decades. For example, it has been studied in terms of the *Vapnik–Chervonenkis (VC) dimension*, which has been introduced in the 1970s in model theory and learning theory [38, 30, 35] and measures the complexity of set systems. In Valiant's framework of *probably approximately correct (PAC) learning* [37], a Boolean concept class is learnable if and only if it has finite VC dimension. Grohe and Turán [22] showed that set systems definable by FO formulas on classes of bounded local clique-width, including classes of bounded genus and classes of bounded degree, have bounded VC dimension. Adler and Adler [1] generalised this to all *nowhere dense* graph classes, a robust notion of sparsity introduced by Nešetřil and Ossona de Mendez [27]. The result of [1] has only recently been generalised by Braunfeld, Dawar, Eleftheriadis, and Papadopoulos to nowhere dense classes of relational structures [10]. While the proofs in [1, 10] rely on non-constructive compactness arguments for first-order logic, Pilipczuk, Siebertz, and Toruńczyk [29] gave a combinatorial proof for the result of [1], yielding a computable upper bound on the VC dimension for FO formulas on effectively nowhere dense graph classes.

With respect to computational complexity, in a seminal result, Grohe, Kreutzer, and Siebertz [19] gave an almost-linear-time algorithm for the FO model-checking problem on nowhere dense graph classes. That is, for every nowhere dense graph class $\mathcal{C}$, they showed that there is an algorithm that checks whether a given FO sentence φ holds in a given graph $G \in \mathcal{C}$ in time $\mathcal{O}(n^{1+\varepsilon})$ for every $\varepsilon > 0$, where n is the number of vertices in the graph, and the constants in the $\mathcal{O}$-notation may depend on $\mathcal{C}$, φ, and ε. This generalises previous results for classes of bounded degree [32], bounded expansion, and locally bounded expansion [14]. Kreutzer and Dawar [24] showed that for classes that are closed under taking subgraphs, the result is optimal, i.e. for every such graph class that is *not* nowhere dense, the FO model-checking problem is just as hard as the model-checking problem on the class of all graphs, which is hard for the parameterised complexity class $\mathsf{AW}[*]$. Schweikardt, Segoufin, and Vigny [31] generalised the result of [19] to first-order testing and enumeration on nowhere dense classes. On a nowhere dense class $\mathcal{C}$ of relational structures, this shows that for every FO formula φ and every relational structure $\mathcal{A} \in \mathcal{C}$, after an almost-linear-time preprocessing step, we can check whether a given tuple of elements satisfies φ on $\mathcal{A}$ in constant time, and we can enumerate all tuples satisfying φ on $\mathcal{A}$ with constant delay.

First-order logic with counting

Such *algorithmic metatheorems*, which state that problems that can be formalised in a certain logic can be solved efficiently on certain classes of inputs, have also been obtained beyond first-order logic. The first-order logic with counting FOC, introduced by Kuske and Schweikardt in [25], extends first-order logic by *counting terms* and *numerical predicates*. Counting terms are polynomials of *#-terms* of the form $\#\bar{x}.\varphi$, which provide a means of counting the number of satisfying assignments of a formula φ. The numerical predicates allow for the comparison of counting terms; that is, for counting terms $t_1, \dots, t_m$ and a numerical predicate $\mathsf{P} \subseteq \mathbb{Z}^m$, $\mathsf{P}(t_1, \dots, t_m)$ is a formula. For example, the numerical predicate $\mathsf{P}_= := \{(i, i) : i \in \mathbb{Z}\}$ facilitates checking that the results of two counting terms are equal. The authors gave algorithms for constant-time testing and constant-delay enumeration for FOC on classes of bounded degree. Beyond classes of bounded degree, Grohe and Schweikardt [21] showed that the model-checking problem for FOC is already $\mathsf{AW}[*]$-hard on the very restrictive class of unranked trees of height 3. The article [21] also defined the fragment FOC_1 of FOC and showed that the model-checking problem for FOC_1 can be solved in almost-linear time on nowhere dense classes. The logic FOC_1 restricts FOC by requiring that for every formula of the form $\mathsf{P}(t_1, \dots, t_m)$, the counting terms $t_1, \dots, t_m$ have only one free variable in total. Generalising the results of [29] from FO to FOC_1, [9] proves a computable upper bound on the VC dimension of FOC_1 formulas on effectively nowhere dense graph classes. Toruńczyk [36] introduced the logic $\mathsf{FO}_G[\mathbb{C}]$, which extends FO by aggregation in semirings. For $\mathsf{FO}_G[\mathbb{C}]$ on classes of bounded expansion, the author gave an algorithm for constant-time query answering after linear-time preprocessing.

Agnostic PAC learning

Valiant's PAC-learning setting mentioned above has been generalised by Haussler [23] to the *agnostic-PAC-learning* setting. In this setting, within the logical framework for Boolean classification problems from [22], we assume a probability distribution $\mathcal{D}$ on $A^\ell \times \{0, 1\}$ for a relational structure $\mathcal{A}$ with universe A and for some $\ell \in \mathbb{N}_{\geq 1}$. Our goal is to find a function $h\colon A^\ell \to \{0, 1\}$, called a *hypothesis*, that (approximately) minimises the probability of making an error on a randomly drawn example, that is, it should minimise $\Pr_{(\bar{w},\lambda)\sim\mathcal{D}}\big(h(\bar{w}) \neq \lambda\big)$.

Hypotheses are represented as pairs $(\varphi, \bar{v})$, where $\varphi(\bar{x}, \bar{y})$ is a formula with $|\bar{y}| = \ell$ and $\bar{v} \in A^{|\bar{x}|}$. This represents the hypothesis $h\colon A^\ell \to \{0,1\}$ with $h(\bar{w}) = 1$ if and only if $\mathcal{A} \models \varphi[\bar{v}, \bar{w}]$. In an agnostic-PAC-learning algorithm, we assume oracle access to $\mathcal{D}$. The algorithm shall draw a few examples from $\mathcal{D}$ and then construct a hypothesis based on these examples. To make the problem feasible at all, one needs to limit the complexity of the hypotheses we are allowed to return; we do this by setting an upper bound on the length of the formula φ, which also upper bounds other relevant complexity measures such as number of variables and quantifier rank. The agnostic-PAC-learning problem for Boolean classification has been studied before for first-order logic on nowhere dense classes [5, 3] and for first-order logic and extensions of first-order logic by counting and weight aggregation on classes of small degree [20, 4, 7]. In [8], the authors generalise the logical framework for Boolean classification problems from [22] to multiclass classification problems, where hypotheses are of the form $h\colon A^\ell \to \mathbb{Z}$, and counting terms from FOC_1 are used to define these hypotheses. However, [8] only considers the so-called *consistent-learning* setting, which is conceptually much simpler than the (agnostic-)PAC-learning setting.

There have been various approaches to generalise the VC dimension to the multiclass setting in order to characterise multiclass learnability. The *Natarajan dimension* and the *graph dimension* were both introduced by Natarajan [26]. As it turns out, neither of them characterise multiclass PAC learnability, since the Natarajan dimension underestimates the complexity (there is a non-learnable concept class of Natarajan dimension 1) and the graph dimension overestimates the complexity (there is a learnable concept class of infinite graph dimension). Only recently, the characterisation of multiclass learnability was resolved by [11] via the *Daniely–Shalev-Shwartz (DS) dimension*, which was introduced in [13]. However, the algorithmic approaches to solve the multiclass agnostic-PAC-learning problem for classes of bounded DS dimension are much more involved than in the Boolean case. Meanwhile, for classes of bounded graph dimension (which implies that they also have bounded DS dimension), the paper [12] shows that one can still use the simpler *empirical-risk-minimisation (ERM)* procedure, which is also used in the Boolean case.

Our contributions

We introduce *clique-guarded first-order logic with counting* (cgFOC), a fragment of FOC that is substantially more expressive than FOC_1. Instead of asking that the counting terms $t_1, \ldots, t_m$ in a formula of the form $\mathsf{P}(t_1, \ldots, t_m)$ have only one free variable in total, we require every pair of free variables of $t_1, \ldots, t_m$ to be guarded by an atom. On graphs, this means that one can compare counting terms in formulas of the form $\bigwedge_{\{x,y\}\in\binom{X}{2}} E(x,y) \wedge \mathsf{P}(t_1, \ldots, t_m)$ for $X \coloneqq \bigcup_{i=1}^m \mathrm{free}(t_i)$, ensuring that the free variables of $t_1, \ldots, t_m$ form a clique in the graph. On relational structures $\mathcal{A}$, formulas in cgFOC ensure that the free variables of the counting terms $t_1, \ldots, t_m$ form a clique in the Gaifman graph $G_{\mathcal{A}}$ of the structure $\mathcal{A}$. The formal definition of the logic cgFOC can be found in Section 3.

In Section 4, as the main result of this paper, we show that counting terms from cgFOC have bounded graph dimension (see Section 4 for the definition) on nowhere dense classes. This result can be viewed as a bound on the expressive power of cgFOC.

▶ **Theorem 1.** *For every (effectively) nowhere dense graph class $\mathcal{C}$, there is a (computable) function $f\colon \mathsf{cgFOC} \to \mathbb{N}$ such that for every cgFOC counting term $t(\bar{x}, \bar{y})$ and every σ-structure $\mathcal{A}$ with $\sigma \supseteq \sigma(t)$ and $G_{\mathcal{A}} \in \mathcal{C}$, the graph dimension of $t(\bar{x}, \bar{y})$ in $\mathcal{A}$ is at most $f(t)$.*

Our proof relies on locality arguments and a lemma that bounds the rank of a "type matrix". To the best of our knowledge, we are the first to study graph dimension in a logical framework.

By the connection between the graph dimension and PAC learning described above, Theorem 1 enables us to solve multiclass PAC-learning problems on sparse classes, where the task is to find a suitable hypothesis h that maps tuples in A^ℓ to values from the *infinite* set $\mathbb{Z}$.

As a direct consequence of Theorem 1, we obtain (computable) upper bounds on the VC dimension of cgFOC formulas on (effectively) nowhere dense classes of relational structures (see Corollary 16). This generalises the computable upper bounds on the VC dimension of [29, 9] from FO and FOC_1 to cgFOC and from classes of graphs to classes of relational structures. In particular, this gives the first *computable* upper bound on the VC dimension of FO formulas on effectively nowhere dense classes of relational structures. This strengthens the result from [10], and we even prove it for the much stronger logic cgFOC.

In addition to the complexity in terms of the graph dimension and the VC dimension, we also study the computational complexity of evaluating cgFOC. On classes of relational structures of locally bounded expansion, we give constant-time query-answering and constant-delay enumeration algorithms for cgFOC with almost-linear-time preprocessing (see Theorem 9). This generalises the model-checking result from [21] for classes of locally bounded expansion from FOC_1 to cgFOC and the query-answering and enumeration results from [33] from FO to cgFOC.

In Section 5, we combine the computable bounds on the graph dimension and the VC dimension with the algorithmic metatheorems for cgFOC to develop agnostic-PAC-learning algorithms for cgFOC-definable concepts. In fact, we even solve a harder problem, *agnostic PAC enumeration*, where we enumerate all hypotheses up to a given complexity bound with constant delay, sorted by their accuracy (descending) and their complexity (ascending). The first hypothesis we output will then be a correct solution for the agnostic-PAC-learning problem. We solve the multiclass agnostic-PAC-enumeration problem on classes of locally bounded expansion for concepts definable via cgFOC counting terms. The algorithm we present (see Theorem 18) preprocesses the input in time polynomial in the size of the relational structure, but the exponent depends on the desired complexity bound of the hypotheses. For classes of bounded degree, we give a different algorithm that has only linear time preprocessing and even works for the full logic FOC. Moreover, for the restriction to Boolean classification, we present an algorithm for the agnostic-PAC-enumeration problem for cgFOC that has almost-linear-time preprocessing on classes of locally bounded expansion.

Finally, in addition to our complexity analysis of cgFOC, we also study variants of this logic. In cgFOC, whenever we want to compare counting terms $t_1, \ldots, t_m$ using a predicate $\mathsf{P} \subseteq \mathbb{Z}^m$, we need to add guards that ensure that the free variables of $t_1, \ldots, t_m$ form a clique, so the free variables shall have pairwise distance at most 1. Slightly weakening these guards leads to a logic that is intractable on very simple graph classes. As we discuss at the end of Section 3, constructs of the form $E(x, y) \wedge E(y, z) \wedge \mathsf{P}_=(t_1(x), t_2(z))$ allow us to interpret the class of all graphs in the class $\mathcal{T}_2$ of coloured trees of height at most 2. Hence, this would give us a formula with unbounded VC dimension and an $\mathsf{AW}[*]$-hard model-checking problem on $\mathcal{T}_2$. Moreover, this would give us a counting term with unbounded graph dimension on $\mathcal{T}_2$, and it is easy to show that it also has unbounded DS dimension and Natarajan dimension.

We also show that even the very restricted fragment of cgFOC with constructs of the form $E(x, y) \wedge \mathsf{P}_=(t_1(x), t_2(y))$ allows us to interpret the class of all graphs in a class of coloured graphs of shrub-depth at most 2. This shows that, while cgFOC is well-suited for sparse classes of relational structures, even much more restricted variants are too strong for very simple dense classes.

2 Preliminaries

We let $\mathbb{Z}$, $\mathbb{N}$, $\mathbb{N}_{\geq 1}$, $\mathbb{Q}$, and $\mathbb{Q}_{>0}$ denote the sets of integers, non-negative integers, positive integers, rationals, and positive rationals, respectively. For $m, n \in \mathbb{Z}$, we let $[m, n] := \{i \in \mathbb{Z} : m \leq i \leq n\}$ and $[n] := [1, n]$. For a k-tuple $\bar{v} = (v_1, \ldots, v_k)$, we write $|\bar{v}|$ to denote its *length* k, and we let $\tilde{v} := \{v_1, \ldots, v_k\}$. For a set X, we let $2^X := \{Y : Y \subseteq X\}$, and for $k \in \mathbb{N}$, we let $\binom{X}{k} := \{Y : Y \subseteq X, |Y| = k\}$.

Graphs and Sparse Graph Classes

When speaking of *graphs*, we mean undirected graphs without self-loops. For a graph G, we write $V(G)$ and $E(G)$ to denote its vertex set and edge set, respectively. For $V' \subseteq V(G)$, we write $G[V']$ to denote the subgraph of G induced by V'.

For two vertices v, w of a graph G, we let $\mathrm{dist}^G(v, w)$, called the *distance* between v and w in G, be the *length* (that is, the number of edges) of a shortest path between v and w. If there is no such path, then $\mathrm{dist}^G(v, w) := \infty$. For $r \in \mathbb{N}$, we let $N_r^G(v) := \{w \in V(G) : \mathrm{dist}^G(v, w) \leq r\}$ be the *r-neighbourhood* of v in G, we let $\mathcal{N}_r^G(v) := G[N_r^G(v)]$ be the corresponding induced subgraph, and we extend this to tuples $\bar{v} \in \big(V(G)\big)^k$ for $k \in \mathbb{N}_{\geq 1}$ by setting $N_r^G(\bar{v}) := \bigcup_{i \in [k]} N_r^G(v_i)$ and $\mathcal{N}_r^G(\bar{v}) := G[N_r^G(\bar{v})]$. We say that a graph G has *radius at most r* if there exists a $v \in V(G)$ such that $G = \mathcal{N}_r^G(v)$. For sets $V, W, S \subseteq V(G)$, we say that *V and W are r-separated by S (in G)* if every path of length $\leq r$ in G from a vertex in V to a vertex in W contains a vertex from S. We use the same notion for tuples $\bar{v}$ and $\bar{w}$ by considering the sets $V = \tilde{v}$ and $W = \tilde{w}$, and we extend the notion to sets of tuples V' and W' by considering the sets $V = \bigcup_{\bar{v} \in V'} \tilde{v}$ and $W = \bigcup_{\bar{w} \in W'} \tilde{w}$. If V and W are r-separated by the empty set, then we say that V and W are r-separated. A *depth-r minor* of G is a subgraph of a graph obtained from G by contracting mutually vertex-disjoint connected subgraphs of radius at most r to single vertices.

▶ **Definition 2.** *Let $\mathcal{C}$ be a class of graphs. We say that $\mathcal{C}$ has* bounded degree *if there is a constant $d \in \mathbb{N}$ such that every graph $G \in \mathcal{C}$ has maximum degree at most d.*

$\mathcal{C}$ has bounded expansion *if there is a function $g \colon \mathbb{N} \to \mathbb{N}$ such that for every $G \in \mathcal{C}$, $r \in \mathbb{N}$, and every depth-r minor H of G, it holds that $\frac{|E(H)|}{|V(H)|} \leq g(r)$.*

$\mathcal{C}$ has locally bounded expansion *if there is a function $g \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for every $d \in \mathbb{N}$, the class of graphs $\mathcal{N}_d^G(v)$ of d-neighbourhoods of vertices $v \in V(G)$ in graphs $G \in \mathcal{C}$ has bounded expansion via the function $g_d \colon \mathbb{N} \to \mathbb{N}, g_d(r) := g(d, r)$. Equivalently, $\mathcal{C}$ has locally bounded expansion if there is a function $g \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for every $G \in \mathcal{C}$, $v \in V(G)$, $d, r \in \mathbb{N}$, and every depth-r minor H of $\mathcal{N}_d^G(v)$, it holds that $\frac{|E(H)|}{|V(H)|} \leq g(d, r)$.*

Lastly, $\mathcal{C}$ is nowhere dense *if there is a function $g \colon \mathbb{N} \to \mathbb{N}$ such that for all $r \in \mathbb{N}$, there is no graph $G \in \mathcal{C}$ that contains the complete graph $K_{g(r)}$ on $g(r)$ vertices as a depth-r minor.*

If the function g above is computable, then we say that $\mathcal{C}$ has effectively bounded expansion*,* effectively locally bounded expansion*, or $\mathcal{C}$ is* effectively nowhere dense*, respectively.*

Every class of bounded degree has effectively bounded expansion, every class of (effectively) bounded expansion has (effectively) locally bounded expansion, and every class of (effectively) locally bounded expansion is (effectively) nowhere dense. We refer to [28] for an in-depth introduction of these notions.

Relational Structures

We use standard notation for structures, and we refer to the full version [6, Appendix A] for details. All structures in this paper are finite, all signatures are relational and finite, and relation symbols R have a non-negative *arity* $\mathrm{ar}(R) \in \mathbb{N}$. Unless explicitly stated otherwise, we use $\mathcal{A}$ and $\mathcal{B}$ and variations such as $\mathcal{A}_1$, $\mathcal{A}'$, etc. to denote structures, we denote their universes by A, B, A_1, A', etc., respectively, and $\sigma(\mathcal{A})$ denotes the signature of $\mathcal{A}$. We let the *representation size* of a structure $\mathcal{A}$ be $\|\mathcal{A}\| \coloneqq |A| + \sum_{R\in\sigma(\mathcal{A})} \max(\mathrm{ar}(R), 1) \cdot |R(\mathcal{A})|$.

For our algorithmic results, we assume a linear order $<$ on the universe A. For example, in our enumeration result, we will output tuples in lexicographic order based on the linear order on A, and we could assume that $A \subset \mathbb{N}$. However, we do not assume that this linear order is realised by some relation symbol $R \in \sigma$. In fact, the classes of structures that we consider in this paper will rule out linear orders of arbitrary size being implemented in σ.

The *Gaifman graph* of a structure $\mathcal{A}$ is the graph $G_{\mathcal{A}}$ with $V(G_{\mathcal{A}}) = A$ and where, for $v, w \in A$, we have $\{v, w\} \in E(G_{\mathcal{A}})$ if and only if $v \neq w$ and $v, w \in \tilde{v}$ for some $\bar{v} \in R(\mathcal{A})$ with $R \in \sigma$. The degree of a structure $\mathcal{A}$ is the degree of its Gaifman graph $G_{\mathcal{A}}$. For a number $r \in \mathbb{N}$ and elements $v, w \in A$, the notions of *distance* $\mathrm{dist}^{\mathcal{A}}(v, w)$ and *neighbourhood* $N_r^{\mathcal{A}}(v)$ naturally extend from graphs to structures by considering the Gaifman graph $G_{\mathcal{A}}$ of $\mathcal{A}$. We write $\mathcal{N}_r^{\mathcal{A}}(v)$ to denote the induced substructure $\mathcal{A}[N_r^{\mathcal{A}}(v)]$ of $\mathcal{A}$ on the set $N_r^{\mathcal{A}}(v)$.

First-Order Logic with Counting

We assume familiarity with first-order logic (FO). Let σ be a signature, and let vars be a fixed and countably infinite set of variables. A *σ-interpretation* $\mathcal{I} = (\mathcal{A}, \beta)$ consists of a σ-structure $\mathcal{A}$ and an *assignment* $\beta\colon \mathsf{vars} \to A$. For $k \in \mathbb{N}$, pairwise distinct variables $x_1, \dots, x_k \in \mathsf{vars}$, and elements $v_1, \dots, v_k \in A$, we write $\mathcal{I}\frac{v_1,\dots,v_k}{x_1,\dots,x_k}$ for the interpretation $(\mathcal{A}, \beta\frac{v_1,\dots,v_k}{x_1,\dots,x_k})$, where $\beta\frac{v_1,\dots,v_k}{x_1,\dots,x_k}$ is the assignment β' with $\beta'(x_i) = v_i$ for every $i \in [k]$ and $\beta'(y) = \beta(y)$ for all $y \in \mathsf{vars} \setminus \{x_1, \dots, x_k\}$.

We consider the logic FOC that Kuske and Schweikardt introduced in [25]. This logic allows building numerical statements based on counting terms as well as numerical predicates. A *numerical predicate collection* is a triple $(\mathbb{P}, \mathrm{ar}, [\![.]\!])$ where $\mathbb{P}$ is a countable set of *predicate names*, and, to each $\mathsf{P} \in \mathbb{P}$, ar assigns an *arity* $\mathrm{ar}(\mathsf{P}) \in \mathbb{N}_{\geq 1}$ and $[\![.]\!]$ assigns a *semantics* $[\![\mathsf{P}]\!] \subseteq \mathbb{Z}^{\mathrm{ar}(\mathsf{P})}$. For the remainder of this paper, fix a numerical predicate collection $(\mathbb{P}, \mathrm{ar}, [\![.]\!])$. In algorithms, we will assume that machines have access to oracles for evaluating the numerical predicates in constant time. That is, given a predicate $\mathsf{P} \in \mathbb{P}$ and a tuple $(i_1, \dots, i_{\mathrm{ar}(\mathsf{P})})$ of integers, answering the oracle call "$(i_1, \dots, i_{\mathrm{ar}(\mathsf{P})}) \in [\![\mathsf{P}]\!]$?" takes time $\mathcal{O}(1)$.

For a signature σ, the set of *formulas* and *counting terms* for $\mathsf{FOC}[\sigma]$ is defined as follows. *Formulas* are built according to the same rules as formulas in $\mathsf{FO}[\sigma]$ (in particular, $R()$ is a formula if $R \in \sigma$ with $\mathrm{ar}(R) = 0$) and the following additional rule:

($\star$) If $\mathsf{P} \in \mathbb{P}$, $m = \mathrm{ar}(\mathsf{P})$ and $t_1, \dots, t_m$ are counting terms, then $\mathsf{P}(t_1, \dots, t_m)$ is a formula.

Counting terms are built according to the following rules: every integer $i \in \mathbb{Z}$ is a counting term; if t_1 and t_2 are counting terms, then $(t_1 + t_2)$ and $(t_1 \cdot t_2)$ are also counting terms; and

(#) If φ is a formula and $\bar{x} = (x_1, \dots, x_k)$ is a tuple of k pairwise distinct variables for some $k \in \mathbb{N}$, then $\#\bar{x}.\varphi$ is a counting term. This includes $k = 0$, so $\#().\varphi$ is a counting term.

Let $\mathcal{I} = (\mathcal{A}, \beta)$ be a σ-interpretation. For a formula φ from $\mathsf{FOC}[\sigma]$, the semantics $[\![\varphi]\!]^{\mathcal{I}}$ is defined in the same way as for $\mathsf{FO}[\sigma]$, extended by

($\star$) $[\![\mathsf{P}(t_1, \dots, t_m)]\!]^{\mathcal{I}} = 1$ if $([\![t_1]\!]^{\mathcal{I}}, \dots, [\![t_m]\!]^{\mathcal{I}}) \in [\![\mathsf{P}]\!]$, and $[\![\mathsf{P}(t_1, \dots, t_m)]\!]^{\mathcal{I}} = 0$ otherwise.

For a counting term t from $\mathsf{FOC}[\sigma]$, the semantics $[\![t]\!]^{\mathcal{I}}$ is defined via $[\![i]\!]^{\mathcal{I}} = i$ for $i \in \mathbb{Z}$; $[\![(t_1 + t_2)]\!]^{\mathcal{I}} = [\![t_1]\!]^{\mathcal{I}} + [\![t_2]\!]^{\mathcal{I}}$ and $[\![(t_1 \cdot t_2)]\!]^{\mathcal{I}} = [\![t_1]\!]^{\mathcal{I}} \cdot [\![t_2]\!]^{\mathcal{I}}$; and

(#) $[\![\#\bar{x}.\varphi]\!]^{\mathcal{I}} = \left|\left\{(v_1, \dots, v_k) \in A^k : [\![\varphi]\!]^{\mathcal{I}\frac{v_1,\dots,v_k}{x_1,\dots,x_k}} = 1\right\}\right|$, where $\bar{x} = (x_1, \dots, x_k)$.

We refer to the full version [6, Definition 21] for a more detailed definition of $\mathsf{FOC}[\sigma]$ that also includes the definition of $\mathsf{FO}[\sigma]$, and we refer to Examples 4 and 5 as well as Remarks 11 and 12 for examples of formulas and counting terms from $\mathsf{FOC}[\sigma]$. For counting terms t_1 and t_2, we write $(t_1 - t_2)$ as a shorthand for $\big(t_1 + ((-1) \cdot t_2)\big)$. Counting terms of the form $\#\bar{x}.\varphi$ are also called *$\#$-terms*. An *expression* is a formula or a counting term. For an expression ξ, we write $\sigma(\xi)$ for the set of all relation symbols that occur in ξ. The *length* $|\xi|$ of an expression ξ is its length when viewed as a word over the alphabet $\sigma \cup \mathsf{vars} \cup \mathbb{P} \cup \mathbb{Z} \cup \{,\} \cup \{=, \neg, \vee, (,), \exists, \#, ., +, \cdot\}$.

The *free variables* $\mathrm{free}(\xi)$ of an expression ξ are inductively defined in the same way as for first-order logic, extended by $\mathrm{free}\big(\mathsf{P}(t_1, \ldots, t_m)\big) = \bigcup_{i=1}^{m} \mathrm{free}(t_i)$; $\mathrm{free}\big(\#(x_1, \ldots, x_k).\varphi\big) = \mathrm{free}(\varphi) \setminus \{x_1, \ldots, x_k\}$; $\mathrm{free}(i) = \emptyset$ for $i \in \mathbb{Z}$; and $\mathrm{free}\big((t_1 + t_2)\big) = \mathrm{free}\big((t_1 \cdot t_2)\big) = \mathrm{free}(t_1) \cup \mathrm{free}(t_2)$. We write $\xi(z_1, \ldots, z_k)$ to indicate that $\mathrm{free}(\xi) \subseteq \{z_1, \ldots, z_k\}$. A *sentence* is a formula without free variables, and a *ground term* is a counting term without free variables.

By FOC, we denote the union of all $\mathsf{FOC}[\sigma]$ for arbitrary signatures σ; this applies analogously to FO.

For a formula φ and a σ-interpretation $\mathcal{I}$, we write $\mathcal{I} \models \varphi$ if $[\![\varphi]\!]^{\mathcal{I}} = 1$. Likewise, $\mathcal{I} \not\models \varphi$ indicates that $[\![\varphi]\!]^{\mathcal{I}} = 0$. For a formula $\varphi(x_1, \ldots, x_k)$, a σ-structure $\mathcal{A}$, and a tuple $\bar{v} = (v_1, \ldots, v_k) \in A^k$, we write $\mathcal{A} \models \varphi[\bar{v}]$ or $(\mathcal{A}, \bar{v}) \models \varphi$ if $(\mathcal{A}, \beta) \models \varphi$ for all assignments β with $\beta(x_i) = v_i$ for all $i \in [k]$. Furthermore, we set $[\![\varphi(\bar{v})]\!]^{\mathcal{A}} := 1$ if $\mathcal{A} \models \varphi[\bar{v}]$, and $[\![\varphi(\bar{v})]\!]^{\mathcal{A}} := 0$ otherwise. If $t(x_1, \ldots, x_k)$ is a counting term, we set $[\![t(\bar{v})]\!]^{\mathcal{A}} := [\![t]\!]^{(\mathcal{A}, \beta)}$ for one (and hence every) assignment β with $\beta(x_i) = v_i$ for all $i \in [k]$. Two expressions ξ, ξ' are *equivalent* if $[\![\xi]\!]^{\mathcal{I}} = [\![\xi']\!]^{\mathcal{I}}$ for all σ-interpretations $\mathcal{I}$. For $d \in \mathbb{N}$, the expressions are called *d-equivalent* if $[\![\xi]\!]^{\mathcal{I}} = [\![\xi']\!]^{\mathcal{I}}$ for all σ-interpretations $\mathcal{I} = (\mathcal{A}, \beta)$ for all structures $\mathcal{A}$ of degree at most d. For an FOC expression $\xi(\bar{x})$ with $|\bar{x}| > 0$ and for an $r \in \mathbb{N}$, we say that $\xi(\bar{x})$ is *r-local* if for every $\sigma(\xi)$-structure $\mathcal{A}$ and every tuple $\bar{v} \in A^{|\bar{x}|}$, it holds that $[\![\xi(\bar{v})]\!]^{\mathcal{A}} = [\![\xi(\bar{v})]\!]^{\mathcal{N}_r^{\mathcal{A}}(\bar{v})}$. We say that $\xi(\bar{x})$ is *local* if it is r-local for some $r \in \mathbb{N}$.

3 Clique-Guarded FOC

In this section, we introduce the logic *clique-guarded first-order logic with counting* (cgFOC). Moreover, we show that cgFOC expressions can be efficiently evaluated on classes of locally bounded expansion, while variants of the logic are already intractable on very simple classes.

For a signature σ, the set of *formulas* and *counting terms* for $\mathsf{cgFOC}[\sigma]$ (clique-guarded first-order logic with counting) is built according to the same rules as formulas and counting terms in $\mathsf{FOC}[\sigma]$, with the following modified version of rule $(\star)$.

$(\star_{\mathsf{cg}})$ If $\mathsf{P} \in \mathbb{P}$, $m = \mathrm{ar}(\mathsf{P})$, $k \in \mathbb{N}$, $R_1, \ldots, R_k \in \sigma$, $\bar{z}_i \in \mathsf{vars}^{\mathrm{ar}(R_i)}$ for all $i \in [k]$, and $t_1, \ldots, t_m$ are counting terms such that for all $z, z' \in \bigcup_{p=1}^{m} \mathrm{free}(t_p)$ with $z \neq z'$, there is some $i \in [k]$ with $z, z' \in \tilde{z}_i$, then $\big(\bigwedge_{i=1}^{k} R_i(\bar{z}_i) \wedge \mathsf{P}(t_1, \ldots, t_m)\big)$ is a formula.

▶ Remark 3. Intuitively, this means that every pair of free variables of a subformula of the form $\mathsf{P}(t_1, \ldots, t_m)$ is guarded by an atom. Hence, the vertices that the free variables of $\mathsf{P}(t_1, \ldots, t_m)$ are interpreted with need to form a clique in the Gaifman graph of the structure. In fact, for every signature σ, every $\mathsf{P} \in \mathbb{P}$, $m := \mathrm{ar}(\mathsf{P})$, and for all $\mathsf{cgFOC}[\sigma]$ counting terms $t_1, \ldots, t_m$, there is a $\mathsf{cgFOC}[\sigma]$ formula φ such that for all σ-interpretations $\mathcal{I} = (\mathcal{A}, \beta)$, it holds that $\mathcal{I} \models \varphi$ if and only if $([\![t_1]\!]^{\mathcal{I}}, \ldots, [\![t_m]\!]^{\mathcal{I}}) \in [\![\mathsf{P}]\!]$ and $\{\beta(x) : x \in \bigcup_{i=1}^{m} \mathrm{free}(t_i)\}$ forms a clique in $G_{\mathcal{A}}$. See the full version [6, Appendix C.1] for such a formula.

▶ **Example 4.** Let $\sigma := \{\bigstar, \hexagon, E, R\}$ for unary relation symbols $\bigstar, \hexagon$, a binary relation symbol E, and a ternary relation symbol R. Furthermore, let $\mathbb{P}$ contain the binary predicate name $\mathsf{P}_=$ with $[\![\mathsf{P}_=]\!] := \{(i, i) : i \in \mathbb{Z}\}$. The $\mathsf{cgFOC}[\sigma]$ counting terms $t_{\bigstar}(x) := \#(z).\big(E(x, z) \wedge \bigstar(z)\big)$

and $t_{\hexagon}(x) \coloneqq \#(z).\big(E(x,z) \wedge \hexagon(z)\big)$ evaluate to the number of $\bigstar$- and $\hexagon$-coloured neighbours of x, respectively. Furthermore, the $\mathsf{cgFOC}[\sigma]$ formula $\varphi(x,y) \coloneqq E(x,y) \wedge \mathsf{P}_{=}\big(t_{\bigstar}(x), t_{\hexagon}(y)\big)$ expresses that x and y are neighbours and the number of $\bigstar$-coloured neighbours of x is equal to the number of $\hexagon$-coloured neighbours of y.

In addition, $u(x) \coloneqq \#(y,z).\big(\bigstar(y) \wedge \hexagon(z) \wedge R(x,y,z)\big)$ is a $\mathsf{cgFOC}[\sigma]$ counting term, and

$$\psi(x_1,\dots,x_6) \coloneqq \Big(R(x_1,x_2,x_4) \wedge R(x_2,x_3,x_5) \wedge R(x_1,x_3,x_6) \wedge \mathsf{P}_{=}\big(u(x_1), u(x_2)+u(x_3)\big)\Big)$$

and $\psi'(x_1,x_2,x_3) \coloneqq \exists x_4 \exists x_5 \exists x_6\, \psi(x_1,\dots,x_6)$ are $\mathsf{cgFOC}[\sigma]$ formulas. The free variables of $\mathsf{P}_{=}\big(u(x_1), u(x_2)+u(x_3)\big)$ are x_1, x_2, x_3. The guards ensure that x_1, x_2, x_3 form a clique in the Gaifman graph. For this, the guards use additional variables x_4, x_5, x_6 that do not need to be contained in any clique.

▶ **Example 5.** We consider a slight variation of the formula $\varphi(x,y)$ from Example 4 in the context of social networks. For this, let $\sigma \coloneqq \{F\}$ be a signature for directed graphs, i.e. F is a binary relation symbol. A σ-structure $\mathcal{A}$ then represents a social network, where the elements of A are the users of the network, and we have $(v,w) \in F(\mathcal{A})$ if the user v is a follower of the user w. Let $\mathbb{P}$ contain the binary predicate name $\mathsf{P}_{>}$ with $[\![\mathsf{P}_{>}]\!] \coloneqq \{(i,j) : i,j \in \mathbb{Z}, i > j\}$. Then

$$\varphi'(x,y) \coloneqq F(x,y) \wedge \mathsf{P}_{>}\big(\#(z).F(z,x), 10 \cdot \#(z).F(z,y)\big)$$

is a $\mathsf{cgFOC}[\sigma]$ formula. It expresses that x is a follower of y and x has more than ten times as many followers as y. Therefore, x may be able to help y increase their popularity.

▶ Remark 6. The logic FOC_1 of [21] is the fragment of FOC where rule $(\star)$ may only be applied if $|\bigcup_{i=1}^{m} \mathrm{free}(t_i)| \leq 1$. It holds that $\mathsf{FOC}_1 \subseteq \mathsf{cgFOC}$. The cgFOC counting terms $t_{\bigstar}$, $t_{\hexagon}$, and u from Example 4 are also FOC_1 counting terms. However, the cgFOC formulas φ, ψ, ψ', and φ' from Examples 4 and 5 are not contained in FOC_1, and there are no FOC_1 formulas equivalent to them.

▶ **Lemma 7.** *For every signature σ and every $\mathsf{cgFOC}[\sigma]$ formula $\varphi(\bar{x})$, there is a signature $\sigma_\varphi \supseteq \sigma$ and an $\mathsf{FO}[\sigma_\varphi]$ formula $\varphi'(\bar{x})$ such that for every σ-structure $\mathcal{A}$, there is a σ_φ-expansion $\mathcal{A}_\varphi$ of $\mathcal{A}$ with $G_{\mathcal{A}_\varphi} = G_{\mathcal{A}}$ such that "$\mathcal{A} \models \varphi[\bar{v}] \iff \mathcal{A}_\varphi \models \varphi'[\bar{v}]$" holds for all $\bar{v} \in A^{|\bar{x}|}$. Moreover, σ_φ, φ' can be computed from σ, φ, and $\mathcal{A}_\varphi$ can be computed from $\varphi, \mathcal{A}$.*

This lemma provides a "computable reduction" from cgFOC formulas to FO formulas; for our purposes it is crucial that the Gaifman graph of the structure remains unchanged. The proof of Lemma 7 is a straightforward structural induction on φ. In this induction, constructs of the form $\mathsf{P}(t_1,\dots,t_m)$ are replaced by fresh relation symbols, and for the corresponding relation we use the set of all tuples that satisfy $\mathsf{P}(t_1,\dots,t_m)$ and that form a clique in the Gaifman graph. The formal proof is given in the full version [6, Appendix C.2]. Enforcing in cgFOC that the free variables in constructs of the form $\mathsf{P}(t_1,\dots,t_m)$ form a clique in the Gaifman graph is essential to obtain an enriched structure $\mathcal{A}_\varphi$ with the same Gaifman graph as $\mathcal{A}$; we will heavily rely on this property throughout the paper. On classes of locally bounded expansion, the computations in Lemma 7 can be done efficiently, as stated in the following result.

▶ **Lemma 8.** *Let $\mathcal{C}$ be a graph class of locally bounded expansion. There is a function f such that the algorithm from Lemma 7 computing $\mathcal{A}_\varphi$ from φ and any $\mathcal{A}$ with $G_{\mathcal{A}} \in \mathcal{C}$, given $\varepsilon \in \mathbb{Q}_{>0}$ as an additional input, can be chosen so that it runs in time $f(\varphi,\sigma,\varepsilon) \cdot |A|^{1+\varepsilon}$.*

We prove this result in the full version [6, Appendix C.4]. Based on Lemma 8, we can prove the following algorithmic metatheorem for cgFOC.

▶ **Theorem 9** (Query Answering and Enumeration). *Let $\mathcal{C}$ be a graph class of locally bounded expansion. There is a function f and an algorithm that does the following. Given a* cgFOC *expression $\xi(\bar{x})$, a σ-structure $\mathcal{A}$ for some $\sigma \supseteq \sigma(\xi)$ with $G_{\mathcal{A}} \in \mathcal{C}$, and given an $\varepsilon \in \mathbb{Q}_{>0}$, after preprocessing in time $f(\xi, \sigma, \varepsilon) \cdot |A|^{1+\varepsilon}$,*

(a) *the algorithm can answer the following queries in time $f(\xi, \sigma, \varepsilon)$: given a tuple $\bar{v} \in A^{|\bar{x}|}$, output $[\![\xi(\bar{v})]\!]^{\mathcal{A}}$, and*
(b) *if ξ is a formula, then the algorithm can enumerate all tuples $\bar{v} \in A^{|\bar{x}|}$ such that $\mathcal{A} \models \xi[\bar{v}]$ with $f(\xi, \sigma, \varepsilon)$ delay in lexicographic order, without duplicates.*

Our main tool in the proof of Lemma 8 is the restriction of Theorem 9a to #-terms of the form $\#\bar{x}.\varphi(\bar{x}, \bar{y})$ for an FO formula φ (see the full version [6, Theorem 31]). This generalises a special case of [36, Theorem 8] from classes of bounded expansion to classes of locally bounded expansion. In addition, we use the following result, which states that the number of cliques of a given size in graphs from a nowhere dense class is small and which we prove in the full version [6, Appendix C.3].

▶ **Lemma 10.** *Let $\mathcal{C}$ be a nowhere dense graph class. There is a function $f\colon \mathbb{N} \times \mathbb{Q}_{>0} \to \mathbb{N}$ such that*

1. *for every $k \in \mathbb{N}_{\geq 1}$, $\varepsilon \in \mathbb{Q}_{>0}$, and $G \in \mathcal{C}$, it holds that G contains at most $f(k, \varepsilon) \cdot |V(G)|^{1+\varepsilon}$ cliques of size k, and*
2. *for every signature σ, every $\varepsilon \in \mathbb{Q}_{>0}$, and every σ-structure $\mathcal{A}$ with $G_{\mathcal{A}} \in \mathcal{C}$, it holds that $\|\mathcal{A}\| \leq |\sigma| \cdot f(k, \varepsilon) \cdot |A|^{1+\varepsilon}$, where $k \in \mathbb{N}$ is the maximum arity of a relation symbol in σ.*

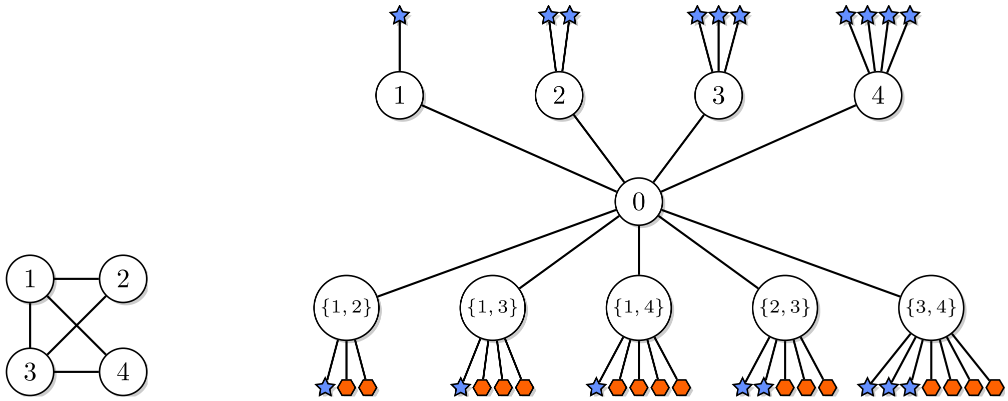
If $\mathcal{C}$ is effectively nowhere dense, then f is computable.

We prove Theorem 9a by combining Lemma 8 with an analogous result for checking whether a tuple satisfies an FO formula in nowhere dense classes in constant time after almost-linear-time preprocessing [31, Corollary 2.4]. Furthermore, we prove Theorem 9b by combining Lemma 8 with the enumeration result [31, Corollary 2.5] for FO in nowhere dense classes. The proof can be found in the full version [6, Appendix C.5].

Theorem 9a generalises the FO model-checking result for classes of locally bounded expansion [14, Corollary 1.2] and the corresponding FO testing result [33, Corollary 15] to the stronger logic cgFOC, and it generalises the FO counting result for classes of locally bounded expansion [33, Theorem 6], to a result on constant-time evaluation of cgFOC counting terms after almost-linear time preprocessing. Theorem 9b generalises the FO enumeration result on classes of locally bounded expansion [33, Corollary 5] to the stronger logic cgFOC.

▶ Remark 11. Rule $(\star_{\text{cg}})$ enforces that vertices interpreting the free variables of $\mathsf{P}(t_1, \ldots, t_m)$ have pairwise distance at most 1 in the Gaifman graph. Weakening this to pairwise distance at most 2 yields a logic that is intractable on very simple classes of graphs. More precisely, as we discuss below, there is a transduction that uses constructs of the form $E(x,y) \wedge E(y,z) \wedge \mathsf{P}_{=}(t_1(x), t_2(z))$ and that produces the class of all graphs from the class of coloured trees of height at most 2, using only two colours. This shows that the model-checking problem on the class of coloured trees of height at most 2 for a logic that allows such constructs is at least as hard as the first-order model-checking problem on the class of all graphs. The latter is known to be $\mathsf{AW}[*]$-hard [16].

We consider the signature $\sigma \coloneqq \{E, ★, ⬢\}$ of coloured graphs, where E is a binary relation symbol, and ★ and ⬢ are unary relation symbols. Let $n \in \mathbb{N}_{\geq 1}$, and let G be a graph with $V(G) = [n]$. We define two formulas φ_V and φ_E and a coloured tree T with colours ★ and ⬢



(a) Input graph G. **(b)** Resulting tree T.

Figure 1 Construction of a coloured tree T based on a graph G such that the transduction (φ_V, φ_E) from Remark 11, applied on T, produces G.

such that φ_V and φ_E do not depend on G or n, we have $V(G) \subseteq V(T)$, and for all $v \in V(T)$, we have $T \models \varphi_V[v]$ if and only if $v \in V(G)$. Moreover, for all $v_1, v_2 \in V(G)$, we will have $T \models \varphi_E[v_1, v_2]$ if and only if $\{v_1, v_2\} \in E(G)$.

See Figure 1 for an example for the construction of the tree T. The tree has a vertex 0 as its root with a child i for every vertex $i \in [n] = V(G)$ and a child e for every edge $e \in E(G)$. In addition, every vertex $i \in V(G)$ has i $\bigstar$-neighbours, and every vertex $\{i, j\} \in E(G) \subseteq V(T)$ with $i < j$ has i $\bigstar$-neighbours and j $\hexagon$-neighbours. Note that T can be computed from G in polynomial time.

The formula $\varphi_V(x)$ then defines the set of vertices $V(G)$ in T by stating that x should have a $\bigstar$-coloured neighbour but no $\hexagon$-coloured neighbour. The formula $\varphi_E(x, y)$ defines the set of edges $E(G)$ in T by stating that x and y form an edge in G if there is a vertex e in T such that the number of $\bigstar$-coloured neighbours of x coincides with the number of $\bigstar$-coloured neighbours of e and the number of $\bigstar$-coloured neighbours of y coincides with the number of $\hexagon$-coloured neighbours of e, or the same holds with reversed roles for x and y. For this, one can use the cgFOC counting terms $t_{\bigstar}(x) := \#(y).\big(E(x, y) \wedge \bigstar(y)\big)$ and $t_{\hexagon}(x) := \#(y).\big(E(x, y) \wedge \hexagon(y)\big)$ to count the number of neighbours of a certain colour. To compare the number of neighbours of two vertices, we can use the fact that the distance between every pair of different vertices from $V(G) \cup E(G) \subset V(T)$ is exactly 2 in T. Hence, for example, to check that the number of $\bigstar$-coloured neighbours of x and e coincide, we can use the formula $\exists z_0\ \big(E(x, z_0) \wedge E(z_0, e) \wedge \mathsf{P}_=\big(t_{\bigstar}(x), t_{\bigstar}(e)\big)\big)$. Note that this formula is an FOC formula, but not a cgFOC formula, since the pair x, e of free variables is not guarded by an edge.

We refer to the full version [6, Appendix C.6] for more details on the construction.

▶ Remark 12. A similar argument shows that even stricter guards than the ones used for rule $(\star_{\text{cg}})$ are not suitable for dense classes: using only constructs of the form $E(x, y) \wedge \mathsf{P}_=(t_1(x), t_2(y))$, that is, where the free variables are guarded by a single edge, we give a transduction in the full version [6, Appendix C.7] that produces the class of all graphs from a class of coloured graphs of shrub-depth at most 2. Hence, the model-checking problem for this restricted logic is already AW[$*$]-hard on a class of coloured graphs of shrub-depth at most 2, which is a very simple dense class [18]. See the full version [6, Appendix C.7] for details.

4 Graph Dimension of cgFOC on Nowhere Dense Classes

In this section, we generalise the results of [29, 9] from FO and FOC_1 on graphs to cgFOC on relational structures. Motivated by the multiclass learning setting as introduced in [8], which we study in Section 5, we also generalise these results from formulas to counting terms, proving that for every nowhere dense class $\mathcal{C}$ and every cgFOC counting term t, the *graph dimension* of t in $\mathcal{C}$, a generalisation of the *VC dimension*, both of which are defined below, is bounded by a value depending only on $\mathcal{C}$ and t. If the class is effectively nowhere dense, the bound will be computable.

The general notions of *VC dimension* and *graph dimension* are defined as follows [38, 26]. Let X be a set. For a set $\mathcal{H} \subseteq 2^X$ and $X' \subseteq X$, we say that $\mathcal{H}$ *shatters* X' if $\{X' \cap H : H \in \mathcal{H}\} = 2^{X'}$. The *VC dimension* of $\mathcal{H}$ is the maximum cardinality of a set $X' \subseteq X$ that is shattered by $\mathcal{H}$. For a set $\mathcal{H}$ of functions $h\colon X \to \mathbb{Z}$ and $X' \subseteq X$, we say that $\mathcal{H}$ *graph shatters* X' if there is a function $g\colon X' \to \mathbb{Z}$ such that for every $Y \subseteq X'$, there is a function $h \in \mathcal{H}$ with $g(x) = h(x)$ for all $x \in Y$ and $g(x) \neq h(x)$ for all $x \in X' \setminus Y$. The *graph dimension* of $\mathcal{H}$ is the maximum cardinality of a set $X' \subseteq X$ that is graph shattered by $\mathcal{H}$.

▶ **Lemma 13.** *Let X be a set, and let $\mathcal{H} \subseteq 2^X$. We let $\mathcal{H}' := \{h_Y : Y \in \mathcal{H}\}$, where, for a set $Y \subseteq X$, we let $h_Y : X \to \mathbb{Z}$ be the indicator function for Y with $h_Y(x) := 1$ if $x \in Y$, and $h_Y(x) := 0$ otherwise.*

A set $X' \subseteq X$ is shattered by $\mathcal{H}$ if and only if it is graph shattered by $\mathcal{H}'$. This implies that the VC dimension of $\mathcal{H}$ is equal to the graph dimension of $\mathcal{H}'$.

Proof. For the forward direction, let $X' \subseteq X$, and let $g\colon X' \to \mathbb{Z}$ with $g(x) = 1$ for all $x \in X'$. If X' is shattered by $\mathcal{H}$, then $\{X' \cap H : H \in \mathcal{H}\} = 2^{X'}$. Hence, for every $Y \subseteq X'$, there is a set $H \in \mathcal{H}$ such that $X' \cap H = Y$. This implies that, for all $x \in X'$, we have $h_H(x) = 1 = g(x)$ if and only if $x \in Y$, and we know that $h_H \in \mathcal{H}'$. Since this holds for every $Y \subseteq X'$, this shows that X' is graph shattered by $\mathcal{H}'$.

Now, for the backward direction, suppose that $X' \subseteq X$ is a set that is graph shattered by $\mathcal{H}'$, and let $g\colon X' \to \mathbb{Z}$ be a witness for this, that is, for every $Y \subseteq X'$, there is a function $h \in \mathcal{H}'$ with $h(x) = g(x)$ for all $x \in Y$ and $h(x) \neq g(x)$ for all $x \in X' \setminus Y$. In particular, there needs to be such a function $h \in \mathcal{H}'$ for $Y = X'$, where $h(x) = g(x)$ for all $x \in X'$. This shows that $g(x) = h(x) \in \{0, 1\}$ for all $x \in X'$. Based on g, we define a mapping $\pi\colon 2^{X'} \to 2^{X'}$, where $\pi(Y) := \{x \in Y : g(x) = 1\} \cup \{x \in X' \setminus Y : g(x) = 0\}$. Now let $Y \subseteq X'$, $Y' := \pi(Y) \subseteq X'$, and let $h \in \mathcal{H}'$ with $h(x) = g(x)$ for all $x \in Y'$ and $h(x) \neq g(x)$ for all $x \in X' \setminus Y'$. Then there is some $H \in \mathcal{H}$ such that $h = h_H$. Let $x \in X'$. If $g(x) = 1$, then $x \in Y$ if and only if $x \in Y'$ if and only if $h(x) = g(x)$ if and only if $h(x) = 1$ if and only if $x \in H$. On the other hand, if $g(x) = 0$, then $x \in Y$ if and only if $x \in X' \setminus Y'$ if and only if $h(x) \neq g(x)$ if and only if $h(x) = 1$ if and only if $x \in H$. Combined, this shows that $H \cap X' = Y$. Since Y was chosen arbitrarily, this shows that X' is shattered by $\mathcal{H}$. ◀

To define the notions of VC dimension and graph dimension for cgFOC, we need the following notation. For a cgFOC formula $\varphi(\bar{x}, \bar{y})$, a σ-structure $\mathcal{A}$ for some $\sigma \supseteq \sigma(\varphi)$, and a set $W \subseteq A$, we let $\mathrm{tp}^{\varphi}_{\mathcal{A}}(\bar{v}/W) := \{\bar{w} \in W^{|\bar{y}|} : \mathcal{A} \models \varphi[\bar{v}, \bar{w}]\}$ for all $\bar{v} \in A^{|\bar{x}|}$. Alternatively, we can view $\mathrm{tp}^{\varphi}_{\mathcal{A}}(\bar{v}/W)$ as a function from $W^{|\bar{y}|}$ to $\{0, 1\}$ that maps a tuple $\bar{w}$ to $[\![\varphi(\bar{v}, \bar{w})]\!]^{\mathcal{A}}$. Accordingly, for a cgFOC counting term $t(\bar{x}, \bar{y})$, we let $\mathrm{tp}^{t}_{\mathcal{A}}(\bar{v}/W)\colon W^{|\bar{y}|} \to \mathbb{Z}$ with $\big(\mathrm{tp}^{t}_{\mathcal{A}}(\bar{v}/W)\big)(\bar{w}) := [\![t(\bar{v}, \bar{w})]\!]^{\mathcal{A}}$ for all $\bar{w} \in W^{|\bar{y}|}$. For a cgFOC expression $\xi(\bar{x}, \bar{y})$ and sets $V, W \subseteq A$, we define $\mathrm{S}^{\xi}_{\mathcal{A}}(V/W) := \{\mathrm{tp}^{\xi}_{\mathcal{A}}(\bar{v}/W) : \bar{v} \in V^{|\bar{x}|}\}$. For a cgFOC counting term $t(\bar{x}, \bar{y})$, a σ-structure $\mathcal{A}$ for some $\sigma \supseteq \sigma(t)$, two sets $V, W \subseteq A$, and $X := W^{|\bar{y}|}$, we study the graph dimension of

$$\mathcal{H} := \mathrm{S}^{t}_{\mathcal{A}}(V/W) = \{\mathrm{tp}^{t}_{\mathcal{A}}(\bar{v}/W) : \bar{v} \in V^{|\bar{x}|}\} = \big\{W^{|\bar{y}|} \to \mathbb{Z}, \bar{w} \mapsto [\![t(\bar{v}, \bar{w})]\!]^{\mathcal{A}} : \bar{v} \in V^{|\bar{x}|}\big\}.$$

We let the *graph dimension of* $t(\bar{x},\bar{y})$ *in* $\mathcal{A}$ be the graph dimension of $\mathrm{S}^t_{\mathcal{A}}(A/A) = \big\{A^{|\bar{y}|} \to \mathbb{Z}, \bar{w} \mapsto [\![t(\bar{v},\bar{w})]\!]^{\mathcal{A}} : \bar{v} \in A^{|\bar{x}|}\big\}$. The *VC dimension* of a cgFOC formula $\varphi(\bar{x},\bar{y})$ in $\mathcal{A}$ is defined as the VC dimension of $\mathrm{S}^{\varphi}_{\mathcal{A}}(A/A) = \big\{\{\bar{w} \in A^{|\bar{y}|} : \mathcal{A} \models \varphi[\bar{v},\bar{w}]\} : \bar{v} \in A^{|\bar{x}|}\big\}$. Next, we prove our main result of this paper, which we repeat for convenience.

▶ **Theorem 1.** *For every (effectively) nowhere dense graph class $\mathcal{C}$, there is a (computable) function $f\colon$ cgFOC $\to \mathbb{N}$ such that for every cgFOC counting term $t(\bar{x},\bar{y})$ and every σ-structure $\mathcal{A}$ with $\sigma \supseteq \sigma(t)$ and $G_{\mathcal{A}} \in \mathcal{C}$, the graph dimension of $t(\bar{x},\bar{y})$ in $\mathcal{A}$ is at most $f(t)$.*

In fact, we prove a stronger result on the *ladder index* of cgFOC counting terms. For a counting term $t(\bar{x},\bar{y})$ and $L \in \mathbb{N}_{\geq 1}$, a *$t$-ladder* of length L in a structure $\mathcal{A}$ is a sequence $\bar{v}_1,\dots,\bar{v}_L,\bar{w}_1,\dots,\bar{w}_L$ such that $\bar{v}_i \in A^{|\bar{x}|}$ and $\bar{w}_i \in A^{|\bar{y}|}$ for all $i \in [L]$ and there is a function $g\colon \{\bar{w}_1,\dots,\bar{w}_L\} \to \mathbb{Z}$ with $[\![t(\bar{v}_i,\bar{w}_j)]\!]^{\mathcal{A}} = g(\bar{w}_j)$ iff $i \leq j$, for all $i,j \in [L]$. The smallest L for which there is no t-ladder of length L in $\mathcal{A}$ is called the *ladder index of* $t(\bar{x},\bar{y})$ *in* $\mathcal{A}$.

▶ **Theorem 14.** *There are computable functions $f\colon$ cgFOC $\times \mathbb{N} \to \mathbb{N}$ and $g\colon$ cgFOC $\to \mathbb{N}$ such that, for every cgFOC counting term $t(\bar{x},\bar{y})$, every $p \in \mathbb{N}$, and every σ-structure $\mathcal{A}$ with $\sigma \supseteq \sigma(t)$ such that $G_{\mathcal{A}}$ excludes K_p as a depth-$g(t)$ minor, the ladder index of $t(\bar{x},\bar{y})$ in $\mathcal{A}$ is at most $f(t,p)$.*

In our proofs, it will be convenient to represent $\mathrm{S}^t_{\mathcal{A}}(V/W)$ as a matrix. For this, we let $M^t_{\mathcal{A}}(V/W) \in \mathbb{Z}^{V^{|\bar{x}|} \times W^{|\bar{y}|}}$ be the matrix with $\big(M^t_{\mathcal{A}}(V/W)\big)_{\bar{v},\bar{w}} \coloneqq [\![t(\bar{v},\bar{w})]\!]^{\mathcal{A}}$. The bound from [29, Corollary 3.2] for formulas would correspond to a bound on the number of different functions in $\mathrm{S}^t_{\mathcal{A}}(V/W)$ or, equivalently, to the number of different rows in $M^t_{\mathcal{A}}(V/W)$. However, in the case of cgFOC counting terms, in general, the number of different functions cannot be constantly bounded. This even holds when replacing all non-zero entries by ones.

In our main technical tool in this section, which is the following generalisation of [29, Corollary 3.2] and [9, Lemma 4.1] for counting terms instead of formulas, we instead prove an upper bound on the *rank* of the matrix $M^t_{\mathcal{A}}(V/W)$ over the reals. Remarkably, this rank bound will then suffice to prove a variety of results concerning the graph dimension and ladder index of cgFOC counting terms on nowhere dense classes.

▶ **Lemma 15.** *There are computable functions $T\colon$ cgFOC $\times \mathbb{N} \to \mathbb{N}$ and $r\colon$ cgFOC $\to \mathbb{N}$ such that, for every cgFOC counting term $t(\bar{x},\bar{y})$, every $m \in \mathbb{N}$, every σ-structure $\mathcal{A}$ for some $\sigma \supseteq \sigma(t)$, and all $V,W \subseteq A$ that are $r(t)$-separated in $G_{\mathcal{A}}$ by a set of size at most m, we have* $\mathrm{rank}\big(M^t_{\mathcal{A}}(V/W)\big) \leq T(t,m)$.

Proof sketch. Every cgFOC counting term is a polynomial of integers and #-terms. For a counting term $t = i$ for an integer i, and for $r(t) \coloneqq 0$, all entries of $M^t_{\mathcal{A}}(V/W)$ are equal to i, so the rank of $M^t_{\mathcal{A}}(V/W)$ is at most 1. For a counting term $t = t_1 + t_2$ or $t = t_1 \cdot t_2$, and for $r(t) \coloneqq \max(r(t_1), r(t_2))$, it can be checked that we have $\mathrm{rank}\big(M^t_{\mathcal{A}}(V/W)\big) \leq \mathrm{rank}\big(M^{t_1}_{\mathcal{A}}(V/W)\big) + \mathrm{rank}\big(M^{t_2}_{\mathcal{A}}(V/W)\big)$ or $\mathrm{rank}\big(M^t_{\mathcal{A}}(V/W)\big) \leq \mathrm{rank}\big(M^{t_1}_{\mathcal{A}}(V/W)\big) \cdot \mathrm{rank}\big(M^{t_2}_{\mathcal{A}}(V/W)\big)$, respectively. Hence, it remains to handle the case of #-terms. For a #-term $t(\bar{x},\bar{y})$ of the form $\#\bar{z}.\varphi(\bar{x},\bar{y},\bar{z})$, by Lemma 7 applied to φ, we can show that it suffices to consider the case where φ is an FO formula. Moreover, by using Gaifman's Theorem [17], we can show that it suffices to consider local formulas φ, and we define $r(t)$ based on the locality radius of φ.

We prove the result by induction on the size m of the set that $r(t)$-separates V and W. In the induction step, we remove one of the separating vertices from the structure and encode the removed information using new relation symbols. This leaves us with the case where V and W are $r(t)$-separated by the empty set.

Since the formula φ is r'-local for some number $r' \in \mathbb{N}$, the evaluation of $[\![\varphi(\bar{v}, \bar{w}, \bar{c})]\!]^{\mathcal{A}}$ for tuples $\bar{v} \in V^{|\bar{x}|}$, $\bar{w} \in W^{|\bar{y}|}$, and $\bar{c} \in A^{|\bar{z}|}$ only depends on the neighbourhood structure $\mathcal{N}_{r'}^{\mathcal{A}}(\bar{v}\bar{w}\bar{c})$. We transform φ into an equivalent formula that is a disjunction of mutually exclusive formulas and that corresponds to a case distinction on which of the vertices from $\bar{v}$, $\bar{w}$, and $\bar{c}$ appear together in the same connected components of $\mathcal{N}_{r'}^{\mathcal{A}}(\bar{v}\bar{w}\bar{c})$.

Since the formulas are mutually exclusive, we can replace the disjunction by a sum of counting terms and then use a Feferman–Vaught decomposition [15] for each of these counting terms to express it as a sum of two-factor products, where each of the factors only depends on the neighbourhood of either V or W. This implies that the matrix $M_{\mathcal{A}}^{t}(V/W)$ can be interpreted as a sum of matrices where the rank of each of the matrices can be bounded by some value that does not depend on $\mathcal{A}$. The formal proof can be found in the full verion [6, Appendix D.1]. ◀

We prove Theorem 14 by assuming the existence of a long t-ladder and showing that this contradicts Lemma 15. Intuitively, starting with a long t-ladder $\bar{v}_1, \ldots, \bar{v}_L, \bar{w}_1, \ldots, \bar{w}_L$, we find a subsequence that is still long, but where every pair of tuples in the sequence is r-separated by the same small set, for a suitable r. Based on this subsequence, we construct r-separated sets V and W, and we argue that the rank of $M_{\mathcal{A}}^{t}(V/W)$ has to be large, which contradicts Lemma 15. We refer to the full version [6, Appendix D.2] for proof details and for the proof of Theorem 1 based on Theorem 14.

For a cgFOC formula $\varphi(\bar{x}, \bar{y})$ and a structure $\mathcal{A}$, by applying Lemma 13, it is easy to show that the VC dimension of $\varphi(\bar{x}, \bar{y})$ in $\mathcal{A}$ is equal to the graph dimension of the counting term $t(\bar{x}, \bar{y}) := \#().\varphi$ in $\mathcal{A}$. Hence, Theorems 1 and 14 imply the following (see the full version [6, Appendix D.3]).

▶ **Corollary 16.** *For every (effectively) nowhere dense graph class $\mathcal{C}$, there is a (computable) function $f\colon$ cgFOC $\to \mathbb{N}$ such that for every cgFOC formula $\varphi(\bar{x}, \bar{y})$ and every σ-structure $\mathcal{A}$ with $\sigma \supseteq \sigma(\varphi)$ and $G_{\mathcal{A}} \in \mathcal{C}$, the VC dimension and the ladder index of $\varphi(\bar{x}, \bar{y})$ in $\mathcal{A}$ are at most $f(\varphi)$.*

In addition, for a cgFOC formula $\varphi(\bar{x}, \bar{y})$, the bound on the rank of $\mathrm{S}_{\mathcal{A}}^{t}(V/W)$ for $t(\bar{x}, \bar{y}) := \#().\varphi$ from Lemma 15 implies a bound on the size of $\mathrm{S}_{\mathcal{A}}^{\varphi}(V/W)$. Based on this, it is straightforward to adapt the proof of [29, Theorem 1.3] to obtain a bound on the *VC density* of cgFOC formulas. The *VC density* of a formula φ on a class of structures $\mathcal{C}$ is the infimum of all reals $\alpha > 0$ such that $\left|\mathrm{S}_{\mathcal{A}}^{\varphi}(A/W)\right| \in \mathcal{O}(|W|^{\alpha})$ for all $\mathcal{A} \in \mathcal{C}$ and all $W \subseteq A$, where constants hidden in the $\mathcal{O}$-notation may depend on α (see, e.g., [2]). In the full version [6, Appendix D.4], we prove the following result:

▶ **Theorem 17.** *Let $\mathcal{C}$ be a nowhere dense graph class, let $\varphi(\bar{x}, \bar{y})$ be a cgFOC formula, let $\sigma \supseteq \sigma(\varphi)$ be a signature, and let $\mathcal{C}'$ be the class of all σ-structures $\mathcal{A}$ with $G_{\mathcal{A}} \in \mathcal{C}$. The VC density of $\varphi(\bar{x}, \bar{y})$ on $\mathcal{C}'$ is at most $|\bar{x}|$.*

5 Learning Concepts Defined by cgFOC

Next, we apply the results of Section 4 to show that concepts that can be defined using cgFOC expressions can be learned efficiently on classes of effectively locally bounded expansion. For this, we consider the logical framework introduced in [22] for Boolean classification problems and its extension to multiclass classification from [8], more specifically, the setting of *agnostic probably approximately correct (PAC) learning* [23], a generalisation of the *PAC-learning* setting [37]. We briefly introduce the considered algorithmic problems here, and we refer to [20, 4] for an in-depth introduction. Let $\mathcal{A}$ be a structure, let $\ell \in \mathbb{N}_{\geq 1}$, and let $\mathcal{D}$ be a probability distribution on $A^\ell \times \mathbb{Z}$. We are given oracle access to $\mathcal{D}$, i.e. each oracle query yields an instance from $A^\ell \times \mathbb{Z}$ drawn from $\mathcal{D}$. We want to find a *hypothesis*, consisting of a counting term $t(\bar{x}, \bar{y})$ with $|\bar{y}| = \ell$ and a tuple $\bar{v} \in A^{|\bar{x}|}$, with small *generalisation error*

$$\mathrm{err}_{\mathcal{D}}(t, \bar{v}) \coloneqq \Pr_{(\bar{w},\lambda)\sim\mathcal{D}}\big(\llbracket t(\bar{v}, \bar{w})\rrbracket^{\mathcal{A}} \neq \lambda\big),$$

that is, with small probability that our hypothesis disagrees with a randomly drawn instance. Since a generalisation error of 0 might not be possible (e.g., when $(\bar{w}, \lambda)$ and $(\bar{w}, \lambda')$ both have probability greater than 0 for some $\bar{w}$ and $\lambda \neq \lambda'$), we want to find a hypothesis with a generalisation error close to the best possible one. In order to make this possible, it is essential to avoid *overfitting*, where the output hypothesis perfectly fits the examples drawn from $\mathcal{D}$, but the algorithm simply memorised the examples instead of learning an underlying rule. Hence, we limit the complexity of the hypotheses we are allowed to return by setting a bound $m \in \mathbb{N}$ on the length of the counting term t, setting a bound on the length k of the tuple $\bar{v}$, and fixing a finite set of integers $I \subset \mathbb{Z}$ that may occur explicitly in t.

Formally, in the *agnostic-PAC-learning problem* for cgFOC, we are given a structure $\mathcal{A}$, $k \in \mathbb{N}$, $\ell, m \in \mathbb{N}_{\geq 1}$, $\varepsilon, \delta \in \mathbb{Q}_{>0}$, a finite set $I \subset \mathbb{Z}$, and oracle access to a probability distribution $\mathcal{D}$ on $A^\ell \times \mathbb{Z}$. We define $T[\sigma, k, \ell, m, I]$ to be the set of all $\mathsf{cgFOC}[\sigma]$ counting terms $t(x_1, \ldots, x_{k'}, y_1, \ldots, y_\ell)$ with $k' \leq k$, $|t| \leq m$ and such that t only uses variables from $\{x_1, \ldots, x_k, y_1, \ldots, y_\ell, z_1, \ldots, z_m\}$ and integers from I. Note that $T[\sigma, k, \ell, m, I]$ is finite and computable from σ, k, ℓ, m, I. An algorithm solving the agnostic-PAC-learning problem may access the oracle to draw examples $(\bar{w}_1, \lambda_1), (\bar{w}_2, \lambda_2), \ldots \in A^\ell \times \mathbb{Z}$ independently and identically distributed (i.i.d.) from $\mathcal{D}$. Let $S = \big((\bar{w}_1, \lambda_1), (\bar{w}_2, \lambda_2), \ldots\big)$ be the sequence of examples drawn by the algorithm. Depending on the drawn sequence S, the algorithm should return a counting term $t_S(\bar{x}, \bar{y}) \in T[\sigma, k, \ell, m, I]$ and a tuple $\bar{v}_S \in A^{|\bar{x}|}$ such that, with probability at least $1 - \delta$ (over the choice of examples drawn i.i.d. from $\mathcal{D}$), it holds that $\mathrm{err}_{\mathcal{D}}(t_S, \bar{v}_S) \leq \varepsilon' + \varepsilon$, where ε' is the minimum generalisation error $\mathrm{err}_{\mathcal{D}}(t, \bar{v})$ of counting terms $t(\bar{x}, \bar{y}) \in T[\sigma, k, \ell, m, I]$ and tuples $\bar{v} \in A^{|\bar{x}|}$.

We also consider a stronger variant of this problem which we call *agnostic PAC enumeration*, where the input is the same as for the learning problem, but instead of asking for one hypothesis that is close to optimal, we ask for an enumeration of *all* hypotheses, sorted by a parameter p that differs from the generalisation error of the hypothesis by at most ε.

The parameter ε, also called the accuracy parameter ("approximately correct"), describes how far the hypothesis we return is allowed to be from an optimal hypothesis. This allows the returned hypothesis to make a few mistakes, for example in case of outliers that are manually handled by an optimal solution but that we do not see in the limited number of examples that we draw in our algorithm. The other parameter δ, also called the confidence parameter ("probably"), describes how confident we are to return a good hypothesis on a randomly drawn sequence of samples. This handles cases where the randomly drawn samples are not representative for $\mathcal{D}$.

As shown in [12, Theorem 5], if a class of hypotheses has bounded graph dimension, then there is a straightforward way to reduce the agnostic-PAC-learning problem to a so-called *empirical-risk-minimisation* (ERM) problem, where the goal is to find a hypothesis with close-to-minimal error on the given sample instead of the entire domain. By combining a refined version of this result with our computable bound on the graph dimension of cgFOC counting terms from Theorem 1, we can show the following.

▶ **Theorem 18.** *For every graph class $\mathcal{C}$ that has effectively locally bounded expansion, there is a function f and an algorithm that does the following. Given a signature σ, a σ-structure $\mathcal{A}$ with $G_{\mathcal{A}} \in \mathcal{C}$, natural numbers $k \in \mathbb{N}$ and $\ell, m \in \mathbb{N}_{\geq 1}$ with $k + \ell \leq m$, rational numbers $\alpha, \varepsilon, \delta \in \mathbb{Q}_{>0}$, a finite set of integers I, and given oracle access to a probability distribution $\mathcal{D}$ on $A^\ell \times \mathbb{Z}$, the algorithm runs in time $f(\alpha, \varepsilon, \delta, \sigma, I, m) \cdot |A|^{\max(k,1+\alpha)}$ and outputs a list consisting of triples $(t, \bar{v}, p)$ in ascending lexicographic order on $(p, |\bar{v}|, |t|)$ such that the list contains exactly one such triple for every $t(\bar{x}, \bar{y}) \in T[\sigma, k, \ell, m, I]$ and every tuple $\bar{v} \in A^{|\bar{x}|}$, and it holds that $p \in \mathbb{Q}$. Moreover, with probability at least $1 - \delta$ over the oracle calls to $\mathcal{D}$, we have that $\left|\mathrm{err}_{\mathcal{D}}(t, \bar{v}) - p\right| \leq \varepsilon$ holds for every triple $(t, \bar{v}, p)$ in the list.*

Our algorithm for Theorem 18 will draw s examples i.i.d. from $\mathcal{D}$ and then run an ERM algorithm on these examples. The main challenge for the proof is to find a suitable number s. While s should be small enough to yield the stated running time, it should also be large enough to ensure that the output of our algorithm is accurate enough with high probability. For this, we use a refinement of [12, Theorem 5], which is analogous to the result on the uniform convergence property in [34, Sections 6.4–6.5]. We refer to the full version [6, Appendix E.1] for details.

Next, we give a much more efficient algorithm on classes of bounded degree that guarantees linear-time preprocessing and constant-delay enumeration. In particular, the exponent in the running time does not depend on k any more. Moreover, the algorithm even works for the much more expressive logic FOC. Analogously to $T[\sigma, k, \ell, m, I]$, we let $T_{\mathsf{FOC}}[\sigma, k, \ell, m, I]$ be the set of all $\mathsf{FOC}[\sigma]$ counting terms $t(x_1, \ldots, x_{k'}, y_1, \ldots, y_\ell)$ with $k' \leq k$, $|t| \leq m$ and such that t only uses variables from $\{x_1, \ldots, x_k, y_1, \ldots, y_\ell, z_1, \ldots, z_m\}$ and integers from I, and this set is also finite and computable from σ, k, ℓ, m, and I.

▶ **Theorem 19.** *If $\mathcal{A}$ is known to have degree at most d, the algorithm from Theorem 18 can be modified to solve agnostic-PAC-enumeration for* FOC *counting terms with delay $f(\varepsilon, \delta, \sigma, I, m, d)$ after preprocessing in time $f(\varepsilon, \delta, \sigma, I, m, d) \cdot |A|$.*

Proof sketch. The full proof can be found in [6, Appendix E.2]. Using the Hanf normal form from [25, Theorem 3.2] for FOC, we can transform an FOC counting term t into a set of cgFOC counting terms, where for every structure $\mathcal{A}$ of degree at most d, one of them is equivalent to t on $\mathcal{A}$. Since Theorem 1 bounds the graph dimension of cgFOC counting terms, FOC counting terms also have bounded graph dimension on classes of bounded degree. Making use of the fact that the graph dimension bounds the number of required examples, we can translate the agnostic-PAC-enumeration problem into a standard enumeration problem for FOC, which can be solved using the result from [25, Corollary 5.7]. ◀

For the last result of this section, we go back to the well-studied *Boolean* agnostic-PAC-learning problem (see, e.g., [20]), i.e. we consider cgFOC formulas instead of counting terms and let the probability distribution $\mathcal{D}$ be on $A^\ell \times \{0, 1\}$, as discussed in Section 1. Again, the exponent in the running time does not depend on k.

▶ **Theorem 20.** *The algorithm from Theorem 18 can be modified so that it solves agnostic-PAC-enumeration for* cgFOC *formulas with delay $f(\alpha, \varepsilon, \delta, \sigma, I, m)$ after preprocessing in time $f(\alpha, \varepsilon, \delta, \sigma, I, m) \cdot |A|^{1+\alpha}$.*

The proof relies on the bound on the VC dimension of cgFOC formulas from Corollary 16 and a translation to a standard enumeration problem, this time for cgFOC formulas, which can be solved using Theorem 9. For details, we refer to the full version [6, Appendix E.3].

6 Final Remarks

The discussion in Remark 11 shows that the logic cgFOC is in some sense optimal, as slightly strengthening the logic makes it intractable on very simple sparse classes. Moreover, the discussion in Remark 12 shows that such guarded logics only work well on sparse classes and are too powerful for dense classes. This, however, does not apply to the logic FOC_1. In fact, it is easy to show that FOC_1 has bounded ladder index or bounded VC dimension on a class if and only if FO has; and this holds for all classes, including dense ones. Meanwhile, the computational complexity of FOC_1 on dense classes remains open.

On sparse classes, we expect the computational complexity of cgFOC and FOC_1 to coincide, and we believe that the techniques in the recently updated full version of [21] for the evaluation of FOC_1 on nowhere dense classes can be generalised to cgFOC. It remains an interesting open question whether one can obtain a multiclass PAC-enumeration algorithm for cgFOC on classes of locally bounded expansion, as in the Boolean case, with almost-linear-time preprocessing and constant delay.

References

1 Hans Adler and Isolde Adler. Interpreting nowhere dense graph classes as a classical notion of model theory. *Eur. J. Comb.*, 36:322–330, 2014. doi:10.1016/j.ejc.2013.06.048.

2 Matthias Aschenbrenner, Alf Dolich, Deirdre Haskell, Dugald Macpherson, and Sergei Starchenko. Vapnik–Chervonenkis density in some theories without the independence property, I. *Transactions of the American Mathematical Society*, 368(8):5889–5949, August 2016. doi:10.1090/tran/6659.

3 Steffen van Bergerem. *Descriptive Complexity of Learning.* PhD thesis, RWTH Aachen University, Germany, 2023. doi:10.18154/RWTH-2023-02554.

4 Steffen van Bergerem. Learning concepts definable in first-order logic with counting. *Log. Methods Comput. Sci.*, 21(3), 2025. doi:10.46298/LMCS-21(3:9)2025.

5 Steffen van Bergerem, Martin Grohe, and Martin Ritzert. On the parameterized complexity of learning first-order logic. In Leonid Libkin and Pablo Barceló, editors, *PODS '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12–17, 2022*, pages 337–346. ACM, 2022. doi:10.1145/3517804.3524151.

6 Steffen van Bergerem, Johannes Friedrich Lange, and Nicole Schweikardt. Complexity of clique-guarded first-order logic with counting, 2026. arXiv:2606.24848v1.

7 Steffen van Bergerem and Nicole Schweikardt. Learning concepts described by weight aggregation logic. In Christel Baier and Jean Goubault-Larrecq, editors, *29th EACSL Annual Conference on Computer Science Logic, CSL 2021, January 25–28, 2021, Ljubljana, Slovenia (Virtual Conference)*, volume 183 of *LIPIcs*, pages 10:1–10:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CSL.2021.10.

8 Steffen van Bergerem and Nicole Schweikardt. Learning aggregate queries defined by first-order logic with counting. In Sudeepa Roy and Ahmet Kara, editors, *28th International Conference on Database Theory, ICDT 2025, March 25–28, 2025, Barcelona, Spain*, volume 328 of *LIPIcs*, pages 4:1–4:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ICDT.2025.4.

9 Steffen van Bergerem and Nicole Schweikardt. On the VC dimension of first-order logic with counting and weight aggregation. In Jörg Endrullis and Sylvain Schmitz, editors, *33rd EACSL Annual Conference on Computer Science Logic, CSL 2025, February 10–14, 2025, Amsterdam, Netherlands*, volume 326 of *LIPIcs*, pages 15:1–15:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CSL.2025.15.

10 Samuel Braunfeld, Anuj Dawar, Ioannis Eleftheriadis, and Aris Papadopoulos. Monadic NIP in monotone classes of relational structures. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming,*

ICALP 2023, July 10–14, 2023, Paderborn, Germany, volume 261 of *LIPIcs*, pages 119:1–119:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.119.

11 Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 – November 3, 2022*, pages 943–955. IEEE, 2022. doi:10.1109/FOCS54457.2022.00093.

12 Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. *J. Mach. Learn. Res.*, 16:2377–2404, 2015. doi:10.5555/2789272.2912074.

13 Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In Maria-Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of the 27th Conference on Learning Theory, COLT 2014, June 13–15, 2014, Barcelona, Spain*, volume 35 of *Proceedings of Machine Learning Research*, pages 287–316. PMLR, 2014. URL: https://proceedings.mlr.press/v35/daniely14b.html.

14 Zdeněk Dvořák, 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.

15 Solomon Feferman and Robert L. Vaught. The first order properties of products of algebraic systems. *Fundamenta Mathematicae*, 47(1):57–103, 1959. URL: http://eudml.org/doc/213526.

16 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.

17 Haim Gaifman. On local and non-local properties. In Jacques Stern, editor, *Proceedings of the Herbrand Symposium*, volume 107 of *Studies in Logic and the Foundations of Mathematics*, pages 105–135. North-Holland Publishing Company, 1982. doi:10.1016/S0049-237X(08)71879-2.

18 Robert Ganian, Petr Hliněný, Jaroslav Nešetřil, Jan Obdržálek, Patrice Ossona de Mendez, and Reshma Ramadurai. When trees grow low: Shrubs and fast MSO_1. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012 – 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27–31, 2012. Proceedings*, volume 7464 of *Lecture Notes in Computer Science*, pages 419–430. Springer, 2012. doi:10.1007/978-3-642-32589-2_38.

19 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.

20 Martin Grohe and Martin Ritzert. Learning first-order definable concepts over structures of small degree. 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.2017.8005080.

21 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 [PODS 2018], Houston, TX, USA, June 10–15, 2018*, pages 253–266. ACM, 2018. Full version on arXiv: https://arxiv.org/abs/1707.05945. doi:10.1145/3196959.3196970.

22 Martin Grohe and György Turán. Learnability and definability in trees and similar structures. *Theory Comput. Syst.*, 37(1):193–220, 2004. doi:10.1007/s00224-003-1112-8.

23 David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992. doi:10.1016/0890-5401(92)90010-D.

24 Stephan Kreutzer and Anuj Dawar. Parameterized complexity of first-order logic. *Electron. Colloquium Comput. Complex.*, TR09-131, 2009. URL: https://eccc.weizmann.ac.il/report/2009/131.

25 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.

26 Balas K. Natarajan. On learning sets and functions. *Mach. Learn.*, 4:67–97, 1989. doi:10.1007/BF00114804.

27 Jaroslav Nešetřil and Patrice Ossona de Mendez. On nowhere dense graphs. *Eur. J. Comb.*, 32(4):600–617, 2011. doi:10.1016/j.ejc.2011.01.006.

28 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.

29 Michał Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. On the number of types in sparse graphs. In Anuj Dawar and Erich Grädel, editors, *33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09–12, 2018*, pages 799–808. ACM, 2018. doi:10.1145/3209108.3209178.

30 Norbert Sauer. On the density of families of sets. *J. Comb. Theory A*, 13(1):145–147, 1972. doi:10.1016/0097-3165(72)90019-2.

31 Nicole Schweikardt, Luc Segoufin, and Alexandre Vigny. Enumeration for FO queries over nowhere dense graphs. *J. ACM*, 69(3):22:1–22:37, 2022. doi:10.1145/3517035.

32 Detlef Seese. Linear time computable problems and first-order descriptions. *Math. Struct. Comput. Sci.*, 6(6):505–526, 1996. doi:10.1017/s0960129500070079.

33 Luc Segoufin and Alexandre Vigny. Constant delay enumeration for FO queries over databases with local bounded expansion. In Michael Benedikt and Giorgio Orsi, editors, *20th International Conference on Database Theory, ICDT 2017, March 21–24, 2017, Venice, Italy*, volume 68 of *LIPIcs*, pages 20:1–20:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICDT.2017.20.

34 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. doi:10.1017/CBO9781107298019.

35 Saharon Shelah. A combinatorial problem: stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972. doi:10.2140/pjm.1972.41.247.

36 Szymon Toruńczyk. Aggregate queries on sparse databases. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14–19, 2020*, pages 427–443. ACM, 2020. doi:10.1145/3375395.3387660.

37 Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984. doi:10.1145/1968.1972.

38 Vladimir N. Vapnik and Alexey Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. doi:10.1137/1116025.