Proceedings Paper:
Lokshtanov, D., Panolan, F., Saurabh, S. et al. (2025) Efficiently Finding and Counting Patterns with Distance Constraints in Sparse Graphs. In: STOC '25: Proceedings of the 57th Annual ACM Symposium on Theory of Computing. The 57th ACM Symposium on Theory of Computing (STOC 2025), 23-27 Jun 2025, Prague, Czech Republic. ACM, New York, NY, United States, pp. 1965-1974. ISBN: 979-8-4007-1510-5.

https://doi.org/10.1145/3717823.3718251

Efficiently Finding and Counting Patterns with Distance Constraints in Sparse Graphs

Daniel Lokshtanov
University of California Santa Barbara
USA
daniello@ucsb.edu

Fahad Panolan
University of Leeds
UK
F.panolan@leeds.ac.uk

Saket Saurabh
The Institute of Mathematical Sciences, HBNI
India
saket@imsc.res.in

Jie Xue
New York University Shanghai
China
jiexue@nyu.edu

Meirav Zehavi
Ben-Gurion University
Israel
meiravze@bgu.ac.il

ABSTRACT

Graph classes of bounded expansion were introduced by Nešetřil and de Mendez as a general model of structurally sparse graphs, which have received considerable attention from both combinatorial and algorithmic perspectives. A celebrated result of Dvořák et al. [JACM'13] showed that any first-order model checking problem on bounded-expansion graph classes is fixed-parameter tractable. A main drawback of the FPT algorithms resulted from this result is the high dependency of their time complexity on the parameter k: the algorithms run in time at least doubly exponential in k, even when the graph class is of polynomial expansion. It is natural to ask whether there exist FPT algorithms for these problem that run in singly exponential time, i.e., $2^{k^{O(1)}} n^{O(1)}$ time.

In this paper, we give a new algorithmic framework for a broad family of first-order model checking problems on sparse graphs, which results in algorithms with running time $2^{k^{O(1)}} \cdot n$ when the graph class is of *exponential expansion* (i.e., the expansion is bounded by a singly exponential function). This covers most well-studied instances of bounded-expansion graph classes, in particular, all polynomial-expansion graph classes. Our framework applies to all problems that can be formulated as finding k vertices in a host graph G with certain *distance constraints*. Furthermore, the framework can be generalized to give $(1 \pm \varepsilon)$-approximation algorithms for the counting versions of these problems with running time $2^{k^{O(1)}} \cdot n(\frac{\log n}{\varepsilon})^{O(1)}$ on exponential-expansion graph classes.

In terms of techniques, our framework differs *entirely* from the one of Dvořák et al. based on centered coloring. We develop various technical components based on the theory of sparse graphs and other tools such as representative sets/functions, tree decomposition, inclusion-exclusion, etc., which are of independent interest.

Remarkably, some of our techniques can be applied to even more general graph classes, such as degenerate graph classes. Therefore, as a byproduct, we obtain a $(1 \pm \varepsilon)$-approximation algorithm for approximately counting bounded-treewidth induced subgraphs in degenerate graphs with running time $k^{O(k)} \cdot (\frac{n}{\varepsilon})^{O(1)}$. This resolves (in a much stronger form) an open problem of Bressan and Roth [FOCS'22], which asked whether such an algorithm exists for counting induced k-matching in degenerate graphs.

CCS CONCEPTS

• **Theory of computation → Parameterized complexity and exact algorithms**; **Graph algorithms analysis**.

KEYWORDS

Bounded expansion graphs, Pattern finding and counting, First-order model checking

ACM Reference Format:
Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. 2025. Efficiently Finding and Counting Patterns with Distance Constraints in Sparse Graphs. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC '25), June 23–27, 2025, Prague, Czechia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3717823.3718251

1 INTRODUCTION

In the seminal work [25, 26], Nešetřil and de Mendez introduced the notion of *bounded-expansion* graph classes as a general model of structurally sparse graphs, which has later received considerable attention from both combinatorial perspective [12, 27, 29, 31, 32] and algorithmic perspective [7–10, 18, 19]. Many well-studied graph classes are of bounded expansion, including proper minor-closed graphs (in particular, planar graphs and bounded-genus graphs), bounded-degree graphs [25], bounded-treewidth graphs, graphs with bounded queue/stack number [29], graphs excluding a fixed topological minor [17], Erdös-Rényi random graphs [29], graphs with sublinear separators [12], various geometric intersection graphs with bounded clique number [11, 23], etc.

While being very general, bounded-expansion graph classes have nice structural properties that are particularly useful in algorithm design, leading to a rich algorithmic theory. Many difficult algorithmic problems on general graphs become substantially more tractable on bounded-expansion graph classes. A typical example

STOC '25, June 23–27, 2025, Prague, Czechia

ACM ISBN 979-8-4007-1510-5/25/06
https://doi.org/10.1145/3717823.3718251

is the Subgraph Isomorphism problem: given a host graph G with n vertices and a pattern graph H with k vertices, find a subgraph of G isomorphic to H. Subgraph Isomorphism has been extensively studied in parameterized complexity and other areas; it generalizes many fundamental problems, such as k-Path, k-Cycle, k-Clique, etc. On general graphs, Subgraph Isomorphism is known to be W[1]-hard [5] (parameterized by k) and cannot be solved in $n^{o(k)}$ time, assuming the Exponential Time Hypothesis (ETH) [14]. In contrast, Nešetřil and de Mendez [26] showed that Subgraph Isomorphism is *fixed-parameter tractable* (FPT) on any graph class of bounded expansion. More generally, a celebrated result of Dvořák et al. [9] gives linear-time FPT algorithms for all first-order model checking problems on bounded-expansion graph classes, parameterized by the length of the input first-order formula.

Although having time complexity linear in n, a main drawback of the FPT algorithms of Dvořák et al. [9] (and also Nešetřil and de Mendez [26]) is the high dependency on the parameter k. These algorithms are based on the technique of *centered coloring* (or *low-treedepth coloring*), which essentially partitions the vertices of the host graph G into $f(k)$ color classes (for some function f) such that any k colors induce a subgraph of treedepth at most k. To resolve Subgraph Isomorphism, one simply guesses the (at most) k colors of G that contain (a copy of) the pattern graph and then applies dynamic programming on the low-treedepth subgraph induced by these colors. Thus, the time complexity of the algorithms heavily relies on the number of color classes, $f(k)$. In the original construction [25], $f(k)$ is at least doubly exponential in k, even if the graph class is of *polynomial expansion* (i.e., the expansion of the graphs in the class is bounded by a polynomial function). Zhu [32] gave an alternative construction of centered coloring, in which $f(k)$ is equal to the 2^{k-2}-th *weak coloring number* of the graph, a sparsity parameter closely related to the expansion of the graph. However, the d-th weak coloring number of a graph G can be exponential in d, even if G is taken from a graph class of polynomial expansion [16], which results in a doubly exponential function $f(k)$ in Zhu's construction. As such, the aforementioned FPT algorithms [9, 26] have time complexity at least doubly exponential in k, even on polynomial-expansion graph classes. This is unsatisfactory, as most FPT problems studied in parameterized complexity can be solved in *singly exponential* time, i.e., $2^{k^{O(1)}}n^{O(1)}$ time. Motivated by this, we ask the following question: can we design $2^{k^{O(1)}}n^{O(1)}$-time algorithms for first-order model checking problems on bounded-expansion graph classes?

If the expansion of the graph class is bounded by an *arbitrary* function $f : \mathbb{N} \to \mathbb{N}$, then the existence of such algorithms seems unlikely, because the k-dependency of the algorithm naturally relies on f, which can grow arbitrarily fast. Therefore, to investigate the above question, some assumption on f is needed. As we are interested in running time singly exponential in k, the most natural assumption here is that f itself is singly exponential, i.e., $f(d) = 2^{d^{O(1)}}$. We say a graph class $\mathcal{G}$ is of *exponential expansion* if its expansion is bounded by a singly exponential function f.

Graph classes of exponential expansion. While there has not been a systematic study on graph classes of exponential expansion, these classes are interesting and natural for a couple of reasons. First, almost all common instances of bounded-expansion graph classes are, in fact, of exponential expansion, including all examples given at the beginning of the paper. Clearly, exponential-expansion graph classes generalize the aforementioned polynomial-expansion graph classes. The latter is an important special case of bounded-expansion graph classes and has been studied intensively in the literature [8, 11–13, 19]. Furthermore, a particularly interesting feature of exponential-expansion graph classes is that these classes can be characterized not only by the expansion function but also by various generalized coloring numbers, which we explain below.

There are three types of generalized coloring numbers widely studied in the literature: admissibility numbers, strong coloring numbers, and the aforementioned weak coloring numbers, introduced by Kierstead and Yang [20] even before bounded-expansion graph classes were introduced. The expansion and these generalized coloring numbers are four different sparsity parameters of a graph class that are closely related to each other. It is well-known that if one of these sparsity parameters is bounded, then all the others are also bounded [28], which characterizes graph classes of bounded expansion. However, the relation among these sparsity parameters is not necessarily polynomial, i.e., one being bounded by a polynomial function does not imply the others are also bounded by polynomial functions. For example, a graph class of polynomial expansion can have exponential weak coloring numbers [16]. Interestingly, if one of these parameters is bounded by a singly exponential function, then all the others are also bounded by singly exponential functions. Thus, graph classes of exponential expansion can be equivalently defined as graph classes with (singly) exponential admissibility numbers, exponential strong coloring numbers, or exponential weak coloring numbers. This makes the notion of exponential-expansion graph classes quite natural and robust.

A general pattern-finding problem. The starting point of this paper is to design $2^{k^{O(1)}}n^{O(1)}$-time algorithms for first-order model checking problems on exponential-expansion graph classes. Previously, Reidl and Sullivan [30] gave an FPT algorithm that can solve Induced Subgraph Isomorphism (and its counting version) in singly-exponential time on exponential-expansion graph classes. Beyond this, however, little was known in this topic. In this paper, we consider a broad family of first-order model checking problems that generalizes (Induced) Subgraph Isomorphism and many other problems, which we call Dist-r Pattern Finding (formally defined below). Roughly speaking, Dist-r Pattern Finding is a generic problem which subsumes all parameterized problems that can be formulated as finding k vertices in a host graph G with certain *distance constraints* that is "bounded" by the number r. We notice that many pattern-finding problems belong to this category. For example, Subgraph Isomorphism can be viewed as finding k vertices in a graph G where the distance between two vertices is equal to 1 if they correspond to neighboring vertices in the pattern graph. Also, Independent Set can be viewed as finding k vertices in a graph G where the distance between two vertices is at least 2.

To define Dist-r Pattern Finding formally, we introduce some notations. By convention, let $[n] = \{1, \dots, n\}$. We say a set $R \subseteq \mathbb{N}_0 = \{0, 1, 2, \dots\}$ is *r-bounded* if either $R \subseteq [r]$ or $\mathbb{N}_0 \backslash R \subseteq [r-1]$. Denote by $\mathcal{R}_r$ the collection of r-bounded sets. For a graph G and a function $D : [k] \times [k] \to \mathcal{R}_r$, a *$D$-pattern* in G is a map $\pi : [k] \to$

$V(G)$ satisfying $\text{dist}_G(\pi(i), \pi(j)) \in D(i, j)$ for all $i, j \in [k]$. Here $\text{dist}_G(u, v)$ denotes the distance between u and v in G. Then Dist-r Pattern Finding is formally defined as follows.

> Dist-r Pattern Finding **Parameter:** k
> **Input:** A graph G of n vertices (in some graph class $\mathcal{G}$) and a function $D : [k] \times [k] \to \mathcal{R}_r$.
> **Goal:** Find a D-pattern in G or conclude its non-existence.

Without loss of generality, one can always assume the input function D satisfies $D(i, i) = \{0\}$ for all $i \in [k]$ and $D(i, j) = D(j, i)$ for all $i, j \in [k]$.

It is easy to see that Subgraph Isomorphism and Graph Homomorphism are instances of Dist-1 Pattern Finding, Induced Subgraph Isomorphism is an instance of Dist-2 Pattern Finding, Distance-r Independent Set is an instance of Dist-r Pattern Finding. Below we explain this in detail. Write $\mathbb{Z}_{\geq t} = \{t, t+1, \dots\}$ for convenience.

- Subgraph Isomorphism. Let H be the pattern graph with $V(H) = \{v_1, \dots, v_k\}$. For $i, j \in [k]$ with $i \neq j$, set $D(i, j) = \{1\}$ if $(v_i, v_j) \in E(H)$ and $D(i, j) = \mathbb{Z}_{\geq 1}$ if $(v_i, v_j) \notin E(H)$. Then the subgraph isomorphisms from H to G are just the D-patterns in G.
- Graph Homomorphism. Let H be the pattern graph with $V(H) = \{v_1, \dots, v_k\}$. For $i, j \in [k]$ with $i \neq j$, set $D(i, j) = \{1\}$ if $(v_i, v_j) \in E(H)$ and $D(i, j) = \mathbb{Z}_{\geq 0}$ if $(v_i, v_j) \notin E(H)$. Then the homomorphisms from H to G are just the D-patterns in G.
- Induced Subgraph Isomorphism. Let H be the pattern graph with $V(H) = \{v_1, \dots, v_k\}$. For $i, j \in [k]$ with $i \neq j$, set $D(i, j) = \{1\}$ if $(v_i, v_j) \in E(H)$ and $D(i, j) = \mathbb{Z}_{\geq 2}$ if $(v_i, v_j) \notin E(H)$. Then the induced subgraph isomorphisms from H to G are just the D-patterns in G.
- Distance-r Independent Set. Set $D(i, j) = \mathbb{Z}_{\geq r}$ for all $i, j \in [k]$ with $i \neq j$. Then the distance-r independent sets in G are just the D-patterns in G.

Counting problems. Besides the pattern-finding problems, we also study their *counting* generalizations, in which the goal is to compute (either exactly or approximately) the number of the corresponding patterns in the host graph. For example, Subgraph Counting, the counting version of Subgraph Isomorphism, asks for the number of subgraph isomorphisms from a pattern graph H to the host graph G. The counting version of all first-order model checking problems (i.e., counting satisfying assignments of first-order formulas) also admits FPT algorithms on graph classes of bounded expansion [28], based on the technique of centered coloring. But again, the time complexity is at least doubly exponential in k, even on polynomial-expansion graphs. Therefore, we also investigate the possibility of designing $2^{k^{O(1)}} \cdot n^{O(1)}$-time algorithms for counting problems on exponential-expansion graph classes. Specifically, we consider the counting version of Dist-r Pattern Finding, which we call Dist-r Pattern Counting.

> Dist-r Pattern Counting **Parameter:** k
> **Input:** A graph G of n vertices (in some graph class $\mathcal{G}$) and a function $D : [k] \times [k] \to \mathcal{R}_r$.
> **Goal:** Compute the number of D-patterns in G.

Clearly, Dist-r Pattern Counting generalizes the counting versions of Subgraph Isomorphism, Induced Subgraph Isomorphism, Graph Homomorphism, Distance-r Independent Set, etc. These problems have been extensively studied on spase graphs in the literature [1–4, 24]. Bressan [2] introduced a novel tree-like decomposition for directed acyclic graphs, and used that to count induced subgraphs of size k in $2^{O(k^2)} \cdot n^{0.25k+2} \log n$ time if the host graph G has bounded degeneracy, and in $2^{O(k^2)} \cdot n^{0.625k+1} \log n$ time if G has bounded average degree. Later Bressan, together with Roth [4] fully classified the complexity of counting subgraphs and induced subgraphs in degenerate host graphs, under the Exponential Time Hypothesis (ETH). In particular, they showed that the tractability of these results is tied with structural properties of the pattern graph, such as the size of a maximum induced matching or a maximum independent set. They also designed algorithms for approximate counting of (induced) subgraphs on degenerate graphs. Finally, Bressan et al. [3] studied counting problems when the host graph G belongs to the family of nowhere dense graphs. One of the main highlights of their work is the characterization of the counting of k-matchings and k-independent sets. In a different direction, Nederlof [24] gave various subexponential parameterized algorithms to count basic patterns such as k-matchings and k-independent sets on planar graphs or, more generally, on apex minor free graphs. This shows that the area of counting patterns in sparse graphs is very active and extremely rewarding.

1.1 Our results

The main contribution of this paper is a new algorithmic framework for Dist-r Pattern Finding on sparse graphs, which results in algorithms with running time $2^{k^{O(1)}} \cdot n$ when applied on graph classes of exponential expansion (in particular, polynomial expansion). The framework can also be generalized to give $(1 \pm \varepsilon)$-approximation algorithms for Dist-r Pattern Counting with running time $2^{k^{O(1)}} \cdot n(\frac{\log n}{\varepsilon})^{O(1)}$ on exponential-expansion graph classes. Interestingly, our approach is entirely different from the existing one based on centered coloring [9, 25, 26]. We shall discuss the techniques we apply in Section 1.2. Before that, we first present the algorithmic results obtained from our framework. Our main result for Dist-r Pattern Finding is the following theorem.

Theorem 1. *Let $r \in \mathbb{N}$ be a number and $\mathcal{G}$ be a graph class of bounded expansion. Then there exists an algorithm for Dist-r Pattern Finding on $\mathcal{G}$ with running time $f(k) \cdot n$ such that*

- *if $\mathcal{G}$ is of exponential expansion, then $f(k) = 2^{k^{O(1)}}$,*
- *if $\mathcal{G}$ is of polynomial expansion, then $f(k) = 2^{O(k^6 \log k)}$.*

When applied to specific instances of Dist-r Pattern Finding, our algorithm is more efficient. For example, for the aforementioned problems, (Induced) Subgraph Isomorphism, Graph Homomorphism, Distance-r Independent Set, our algorithm runs in $2^{O(k^3 \log k)} \cdot n$ on polynomial-expansion graphs.

Next, we discuss our results for Dist-r Pattern Counting. Applying the same algorithmic framework as in Theorem **??**, we give $(1 \pm \varepsilon)$-approximation algorithms for Dist-r Pattern Counting with slightly worse bounds.

Theorem 2. *Let $r \in \mathbb{N}$ be a number and $\mathcal{G}$ be a graph class of bounded expansion. Then there exists a randomized $(1 \pm \varepsilon)$-approximation algorithm for Dist-r Pattern Counting on $\mathcal{G}$ with running time $f(k) \cdot n(\frac{\log^5 n}{\varepsilon^{12}})$ such that*

- *if $\mathcal{G}$ is of exponential expansion, then $f(k) = 2^{k^{O(1)}}$,*
- *if $\mathcal{G}$ is of polynomial expansion, then $f(k) = 2^{O(k^{10} \log k)}$.*

The algorithm has a success probability at least $1 - \frac{1}{n^{ck}}$ for any constant $c > 0$.

The randomness of the algorithm in the above theorem comes from the computation of representative functions [21, 22], and thus it could be derandomized directly if a deterministic construction of representative functions was given.

Interestingly, the techniques used in our framework are not restricted to bounded-expansion graphs. Some of them can be indeed extended to even more general graph classes, such as *degenerate* graph classes. Remarkably, we achieve the following result, as a byproduct, for approximately counting induced subgraphs in degenerate graphs when the pattern graph has bounded treewidth.

Theorem 3. *There exists a randomized $(1 \pm \varepsilon)$-approximation algorithm for Induced Subgraph Counting on degenerate graph classes with running time $k^{O(k)} \cdot n^{\mathrm{tw}(H)+1}(\frac{\log^5 n}{\varepsilon^{12}})$. The algorithm has a success probability $1 - \frac{1}{n^{ck}}$ for any constant $c > 0$.*

The above theorem resolves (in a much stronger form) an open problem of Bressan and Roth [4], which asked whether counting induced k-matching in d-degenerate graphs admits $(1 \pm \varepsilon)$-approximation (possibly randomized) algorithms with running time $f(k,d) \cdot (\frac{n}{\varepsilon})^{O(1)}$. Exact algorithm for this problem with running time $f(k,d) \cdot n^{o(k/\log k)}$ cannot exist, unless ETH fails [4].

1.2 Our techniques

As aforementioned, our algorithmic framework differs entirely from the one based on centered coloring [9, 25, 26]. Instead, it is built on four technical components developed in this paper, each of which is of independent interest. To discuss these techniques, we need to first briefly review the notion of *strong/weakly coloring numbers*. Consider a graph G and an ordering σ of $V(G)$. For $u, v \in V(G)$, we say u is *strongly reachable* (resp., *weakly reachable*) from v within ℓ steps under σ if $u \geq_\sigma v$ and there exists a path in G between u and v of length at most ℓ whose internal vertices are all smaller than v (resp., u) under σ. Denote by $\mathrm{SR}_\ell(G, \sigma, v)$ (resp., $\mathrm{WR}_\ell(G, \sigma, v)$) the set of all vertices of G strongly (resp., weakly) reachable from v within ℓ steps under σ. Define $\mathrm{scol}_\ell(G, \sigma) = \max_{v \in V(G)} |\mathrm{SR}_\ell(G, \sigma, v)|$ and $\mathrm{wcol}_\ell(G, \sigma) = \max_{v \in V(G)} |\mathrm{WR}_\ell(G, \sigma, v)|$. The ℓ-th *strong coloring number* (resp., *weak coloring number*) of G, denoted by $\mathrm{scol}_\ell(G)$ (resp., $\mathrm{wcol}_\ell(G)$), is the minimum of $\mathrm{scol}_\ell(G, \sigma)$ (resp., $\mathrm{wcol}_\ell(G, \sigma)$) over all orderings σ of $V(G)$. For a graph class $\mathcal{G}$, write $\mathrm{scol}_\ell(\mathcal{G}) = \sup_{G \in \mathcal{G}} \mathrm{scol}_\ell(G)$ and $\mathrm{wcol}_\ell(\mathcal{G}) = \sup_{G \in \mathcal{G}} \mathrm{wcol}_\ell(G)$.

Ordering algorithm. Our main algorithms rely on an ordering σ of the vertices of the host graph G that approximates the k-th strongly coloring numbers of G. Dvořák [6] gave two algorithms for computing such an ordering. The first one runs in $O(n^4)$ time. The second one runs in $f(k) \cdot n$ time, where $f(k)$ is at least doubly exponential in k, since it uses the FPT algorithm [9] for first-order model checking on bounded-expansion graphs. Both algorithms are not satisfying for our purpose, since we seek algorithms with running time singly exponential in k and linear (or near-linear) in n. Our first technical component is a new ordering algorithm, which takes an n-vertex graph G as input and computes in $\mathrm{scol}_k^{O(k)}(G) \cdot n$ time an ordering σ of $V(G)$ that satisfies $\mathrm{scol}_k(G, \sigma) = \mathrm{scol}_k^{O(k)}(G)$.

New approach for subgraph isomorphism. Our second technical component is a new approach for Subgraph Isomorphism (and its counting version) on bounded-expansion graphs, which does not rely on centered coloring at all. Our approach combines the tools of representative sets and representative functions with the small strong coloring numbers of the host graph G in a novel way. As a key ingredient of this approach, we show that given an ordering σ of $V(G)$, for any subgraph isomorphism $\pi : V(H) \rightarrow V(G)$ from H to G, there always exists a tree decomposition $\mathcal{T}$ of H satisfying the following (among other nice properties): π maps each bag of $\mathcal{T}$ to a set of vertices in G that are strongly reachable (under the ordering σ) from a single vertex in G within k steps. This result essentially allows us to efficiently find/count subgraph isomorphisms from H to G by first guessing $\mathcal{T}$ and then applying a bottom-up DP-type algorithm on $\mathcal{T}$ together with representative sets/functions. With a careful implementation, we can solve Subgraph Isomorphism in $\mathrm{scol}_k^{O(k)}(G, \sigma) \cdot n$ time and (approximate) Subgraph Counting in $\mathrm{scol}_k^{O(k)}(G, \sigma) \cdot n(\frac{\log^5 n}{\varepsilon^{12}})$ time, provided the ordering σ.

If we use our ordering algorithm to compute an ordering σ satisfying $\mathrm{scol}_k(G, \sigma) = \mathrm{scol}_k^{O(k)}(G)$ and apply the above approach with σ to solve Subgraph Isomorphism/Counting, then the k-dependency of the resulting running time is $\mathrm{scol}_k^{O(k^2)}(G)$. When G is taken from a graph class of exponential expansion, $\mathrm{scol}_k(G) = 2^{k^{O(1)}}$ and thus the running time is singly exponential in k. This proves our results for the special case of Subgraph Isomorphism (and Subgraph Counting). However, to go beyond Subgraph Isomorphism, additional techniques are required.

Distance-based representative sets/functions. Our third technical component is a generalized version of representative sets and representative functions. To introduce this notion, we need to briefly review representative sets/functions. Let U be a universe set, and $\mathcal{P}$ be a collection of subsets of U. A *q-representative set* of $\mathcal{P}$ is a sub-collection $\mathcal{P}' \subseteq \mathcal{P}$ satisfying the following: for any $Q \subseteq U$ with $|Q| = q$, if there exists $P \in \mathcal{P}$ such that $P \cap Q = \emptyset$, then there also exists $P' \in \mathcal{P}'$ such that $P' \cap Q = \emptyset$. The notion of representative function is the counterpart of representative set used for approximate counting (we omit its formal definition here as it is technical). Now consider the setting where $U = V(G)$ for a graph G. We say two subsets $P, Q \subseteq V(G)$ are *r-far* in G if $\mathrm{dist}_G(u,v) \geq r$ for all $u \in P$ and $v \in Q$. Clearly, $P \cap Q = \emptyset$ iff P and Q are 1-far in G. As such, we can generalize the standard representative sets as follows. For a collection $\mathcal{P}$ of subsets of U, a *distance-r q-representative set* of $\mathcal{P}$ is a sub-collection $\mathcal{P}' \subseteq \mathcal{P}$ satisfying the following: for any $Q \subseteq U$ with $|Q| = q$, if there exists $P \in \mathcal{P}$ such that P and Q are r-far in G, then there also exists $P' \in \mathcal{P}'$ such that P' and Q are r-far in G. We call this generalization *distance-based representative sets* for convenience. In the same way, we can also generalize representative functions to distance-based ones. (In fact, our actual definition of distance-based representative sets/functions is even more general, which allows different distance lower bounds for different pairs of vertices in P and Q. Here, we only discuss this simplified version for convenience.)

It is easy to see that for a general graph G, one cannot hope for the existence of small distance-based representative sets/functions for any $r \geq 2$ (we shall give such a hard example in Section 2.3). Interestingly, we show that for sparse graphs, small distance-based representative sets/functions exist and can be computed efficiently. More precisely, if $r \in \mathbb{N}$ is a fixed number and G is a graph whose $(r-1)$-th weakly coloring number is bounded by a constant, then for any collection $\mathcal{P}$ of size-p subsets of $V(G)$, a distance-r q-representative set $\mathcal{P}' \subseteq \mathcal{P}$ with $|\mathcal{P}'| = k^{O(k)}$ can be efficiently computed, where $k = p + q$. For distance-based representative functions, we obtain similar results with slightly worse bounds. Note that here the graphs our results apply to is more general than bounded-expansion ones, as we only require the $(r-1)$-th weakly coloring number to be bounded. Remarkably, our result can construct distance-2 representative sets/functions in all graphs whose first weakly coloring number is bounded, which are exactly degenerate graphs. This explains why our algorithm in Theorem 3 works for any degenerate graph class.

Replacing the (standard) representative sets/functions in our approach for Subgraph Isomorphism/Counting discussed above with the distance-based ones, the approach can be extended to solve most interesting instances of Dist-r Pattern Finding/Counting. Indeed, combining the above three technical components (together with some additional ideas), we can already prove our main results Theorem 1 and Theorem 3. To obtain Theorem 2, the algorithm for Dist-r Pattern Counting, we need an additional building block.

Efficient inclusion-exclusion. Our last technical component is an efficient counting scheme that combines inclusion-exclusion principle with the sparsity of the host graph. This counting scheme essentially reduces the task of counting D-patterns in a graph G (for a given function $D : [k] \times [k] \to [r]_0$) to the task of counting homomorphisms from a small pattern graph to G that satisfy certain distance lower bound constraints. The latter can be further reduced easily to counting subgraph isomorphisms (with distance lower bounds), which can be solved using our techniques discussed above. The counting scheme is rather technical. The basic idea is to use certain homomorphisms to *witness* the D-patterns in G and apply inclusion-exclusion with those homomorphisms. Exploiting the sparsity of G, we are able to convert the inclusion-exclusion formula to an equivalent but much more tractable formula, where the latter can be computed via counting homomorphisms to G with distance lower bounds. The reduction procedure can be done in time singly exponential in k.

We believe that our techniques discussed above can find further applications in other problems, especially the ordering algorithm and the distance-based representative sets/functions.

2 OVERVIEW

Due to the limited space, in this version of the paper, we only provide an overview of our approach, specifically, the first three technical components mentioned in Section 1.2, i.e., the ordering algorithm, the approach for subgraph isomorphism and the distance-based representative sets/functions. The detailed proofs of the main theorems will be deferred to the full version of the paper.

2.1 Ordering algorithm

We first overview our efficient algorithm for approximating the strongly coloring number (and thus the other related sparsity parameters) of a graph. The result we have is the following.

Theorem 4. *Given a graph G with n vertices and a number $k \in \mathbb{N}$, one can compute an ordering σ of $V(G)$ in $\mathrm{scol}_k^{O(k)}(G) \cdot n$ time such that $\mathrm{scol}_k(G,\sigma) = \mathrm{scol}_k^{O(k)}(G)$.*

To discuss the algorithm, we introduce some basic notions. For an integer $p \in \mathbb{N}$, the *Catalan sequence* starting with p is a sequence $C_1^{(p)}, C_2^{(p)}, \dots$ where $C_1^{(p)} = p$ and $C_i^{(p)} = \sum_{j=1}^{i-1} C_j^{(p)} \cdot C_{i-j}^{(p)}$ for $i \geq 2$. By induction, it is easy to see that $C_i^{(p)} = \frac{1}{i}\binom{2i-2}{i} \cdot p^i = (2p)^{O(i)}$. For a set $A \subseteq V(G)$, a vertex $v \in V(G)\backslash A$, and a number $\ell \in \mathbb{N}_0$, we denote by $\Gamma_A(v,\ell)$ the set of all vertices $v' \in V(G)\backslash A$ such that there exists a path in G between v and v' of length at most ℓ whose internal vertices are all in A.

Our ordering algorithm itself is simple, which is shown in Algorithm 1, while the analysis and the linear-time implementation are nontrivial. The algorithm starts with $p = 1$. In each round of the repeat-loop (line 2-12), it tries to construct the ordering using the current p. The variable f is a flag indicating whether the construction is successful or not. If the construction succeeds (f = true), then the algorithm terminates and returns the ordering constructed (line 13). If the construction fails (f = false), then we increase p by 1 (line 11) and proceed to the next round. For a specific p, the construction of the ordering is done by the for-loop (line 4-10). In the i-th iteration of the for-loop, we determine the i-th vertex v_i in the ordering. The set A consists of the vertices that are already in the sequence, i.e., $A = \{v_1, \dots, v_{i-1}\}$. We simply find an arbitrary vertex $v \in V(G)\backslash A$ satisfying $|\Gamma_A(v,\ell)| \leq C_\ell^{(p)}$ for all $\ell \in [k]$ (line 5), and then set $v_i = v$ (line 6). If such a vertex does not exist, then the construction of the ordering for the current p fails; in this case, we set f = false and break from the for-loop.

It is obvious that Algorithm 1 always terminates, since when p is sufficiently large (say, $p = n$), the ordering can be constructed successfully. To guarantee that the ordering approximates the strongly coloring number of G, we need to show that the algorithm terminates with a much smaller p. To this end, we first establish a good property of the Catalan sequences.

For a non-empty sequence $L = (a_1, \dots, a_c)$ of positive integers, we write $\mathrm{sum}(L) = \sum_{i=1}^{c} a_i$ and $\Pi_L^{(p)} = \prod_{i=1}^{c} C_{a_i}^{(p)}$. For a number

Algorithm 1 ORDER(G, k)

1: $n \leftarrow |V(G)|$ and $p \leftarrow 1$
2: **repeat**
3: $\quad A \leftarrow \emptyset$ and $f \leftarrow$ true
4: $\quad$**for** $i = 1, \dots, n$ **do**
5: $\qquad$**if** there exists $v \in V(G)\backslash A$ such that $|\Gamma_A(v, \ell)| \leq C_\ell^{(p)}$ for all $\ell \in [k]$ **then**
6: $\qquad\quad v_i \leftarrow v$
7: $\qquad\quad A \leftarrow A \cup \{v\}$
8: $\qquad$**else**
9: $\qquad\quad f \leftarrow$ false
10: $\qquad\quad$**break**
11: $\quad p \leftarrow p + 1$
12: **until** $f =$ true
13: **return** $(v_1, \dots, v_n)$

$i \in \mathbb{N}$, denote by $\mathcal{L}_{<i}$ the set of all (nonempty) sequences L of positive integers with $\mathsf{sum}(L) < i$. The Catalan sequences have the following property.

Fact 5. *For any $p, i \in \mathbb{N}$, we have $C_i^{(p)} \geq p + \sum_{L \in \mathcal{L}_{<i}} \Pi_L^{(p)} \cdot (p-1)$.*

Using the above fact, we show that Algorithm 1 terminates in at most $\mathsf{scol}_k(G)$ rounds.

LEMMA 6. *When Algorithm 1 terminates, $p \leq \mathsf{scol}_k(G)$.*

PROOF. We shall show that when $p = \mathsf{scol}_k(G)$, the for-loop in line 4-10 of Algorithm 1 can successfully construct all of $v_1, \dots, v_n$, and hence Algorithm 1 always terminates with $p \leq \mathsf{scol}_k(G)$. To this end, we fix an (unknown) ordering σ of $V(G)$ such that $\mathsf{scol}_k(G, \sigma) = \mathsf{scol}_k(G)$.

Let $p = \mathsf{scol}_k(G)$. Consider the i-th iteration of the for-loop. At this point, $v_1, \dots, v_{i-1}$ have already been constructed. Set $A_j = \{v_1, \dots, v_{j-1}\}$ for $j \in [i]$. By the algorithm, for each $j \in [i]$, we have $|\Gamma_{A_j}(v_j, \ell)| \leq C_\ell^{(p)}$ for all $\ell \in [k]$. It suffices to show that the algorithm can successfully construct v_i and proceed to the $(i+1)$-th iteration. Equivalently, we want to prove the existence of a vertex $v \in V(G)\backslash A_i$ satisfying $|\Gamma_{A_i}(v, \ell)| \leq C_\ell^{(p)}$ for all $\ell \in [k]$.

Indeed, we show that if v is the smallest vertex in $V(G)\backslash A_i$ under the ordering σ, then $|\Gamma_{A_i}(v, \ell)| \leq C_\ell^{(p)}$ for all $\ell \in [k]$. This is nontrivial and we achieve it via several steps. Fix a number $\ell \in [k]$. Let $L = (a_1, \dots, a_c) \in \mathcal{L}_{<\ell}$. We say a sequence $(v_{i_1}, \dots, v_{i_c}, v')$ of vertices in G is an *L-type sequence* if the following conditions hold:

- $i_1, \dots, i_c \in [i-1]$ and $i_1 < \cdots < i_c$,
- $v_{i_1} \in \mathsf{SR}_k(G, \sigma, v)$,
- $v_{i_{t+1}} \in \Gamma_{A_{i_t}}(v_{i_t}, a_t)$ for $t \in [c-1]$ and $v' \in \Gamma_{A_{i_c}}(v_{i_c}, a_c)$.

Let $\#_L$ be the number of different L-type sequences. We claim that $\#_L \leq \Pi_L^{(p)} \cdot (p-1)$. Regarding the L-type sequences as strings of length $c+1$, we consider the trie (or prefix tree) T built on these strings, which is a tree of depth $c+1$. The root-leaf paths in T one-to-one correspond to the L-type sequences, and thus $\#_L$ is equal to the number of leaves of T. Since any L-type sequence $(v_{i_1}, \dots, v_{i_c}, v')$ satisfies $v_{i_1} \in \mathsf{SR}_k(G, \sigma, v)$ and $v_{i_1} \neq v$, the degree of the root of T is at most $|\mathsf{SR}_k(G, \sigma, v)\backslash\{v\}| \leq \mathsf{scol}_k(G, \sigma) - 1 = \mathsf{scol}_k(G) - 1 = p - 1$. Now consider a node x of T at level $t \in [c]$ which corresponds to the prefix $(v_{i_1}, \dots, v_{i_t})$ of some L-type sequence. By definition, in any L-type sequence with prefix $(v_{i_1}, \dots, v_{i_t})$, the successor of v_{i_t}, is contained in $\Gamma_{A_{i_t}}(v_{i_t}, a_t)$. Thus, the degree of x in T is at most $|\Gamma_{A_{i_t}}(v_{i_t}, a_t)| \leq C_{a_t}^{(p)}$. It follows that the number of leaves in T (i.e., $\#_L$) is at most $(p-1) \cdot \prod_{t=1}^{c} C_{a_t}^{(p)} = \Pi_L^{(p)} \cdot (p-1)$.

Next, we prove that for each $v' \in \Gamma_{A_i}(v, \ell)$, either $v' \in \mathsf{SR}_\ell(G, \sigma, v)$ or there exists an L-type sequence ending at v' for some $L \in \mathcal{L}_{<\ell}$. Consider a vertex $v' \in \Gamma_{A_i}(v, \ell)$ such that $v' \notin \mathsf{SR}_\ell(G, \sigma, v)$. We are going to construct an L-type sequence ending at v' for some $L \in \mathcal{L}_{<\ell}$. As $v' \in \Gamma_{A_i}(v, \ell)$, there exists a path π in G from v to v' of length at most ℓ whose internal vertices are all in A_i. Without loss of generality, we may assume π is simple. Note that v' is greater than or equal to v under σ, since v is the smallest vertex in $V(G)\backslash A_i$. Thus, there is at least one internal vertex of π greater than v under σ, for otherwise we have $v' \in \mathsf{SR}_\ell(G, \sigma, v)$ because of the existence of π. For two (different) vertices u, u' on π, we write $u <_\pi u'$ if u appears before u' on π (that is, u is closer to the v-end of π than u'). We say a sequence $(i_1, \dots, i_c)$ of numbers in $[i-1]$ is *good* if it satisfies

- $i_1 < \cdots < i_c$,
- $v_{i_1}, \dots, v_{i_c}$ are internal vertices on π and $v_{i_1} \prec_\pi \cdots \prec_\pi v_{i_c}$,
- $v_{i_1} \in \mathsf{SR}_k(G, \sigma, v)$,
- for $t \in [c-1]$, any vertex u on π with $v_{i_t} \prec_\pi u \prec_\pi v_{i_{t+1}}$ is contained in A_{i_t}.

Let c be the largest integer such that there exists a good sequence of length c. We observe that $c \geq 1$. Recall that at least one internal vertex of π is greater than v (under σ). Let z be the internal vertex of π greater than v satisfying that z' is smaller than v for all $z' \prec_\pi z$ (i.e., z is the first vertex on π greater than v). As $z \in A_i$ (for the internal vertices of π all belong to A_i), we have $z = v_j$ for some $j \in [i-1]$. Furthermore, by the choice of z, the sub-path of π between v and z witnesses that $z \in \mathsf{SR}_\ell(G, \sigma, v) \subseteq \mathsf{SR}_k(G, \sigma, v)$. Thus, (j) is a good sequence, which implies $c \geq 1$.

Now let $(i_1, \dots, i_c)$ be a good sequence. By definition, $i_1 < \cdots < i_c$ and $v_{i_1} \prec_\pi \cdots \prec_\pi v_{i_c}$. For $t \in [c-1]$, define π_t as the subpath of π between v_{i_t} and $v_{i_{t+1}}$. Also, define π_c as the subpath of π between v_{i_c} and v'. We claim that $(v_{i_1}, \dots, v_{i_c}, v')$ is an L-type sequence for $L = (a_1, \dots, a_c)$, where a_t is the length of π_t for $t \in [c]$. By definition, we already have $i_1 < \cdots < i_c$ and $v_{i_1} \in \mathsf{SR}_k(G, \sigma, v)$. So it suffices to show $v_{i_{t+1}} \in \Gamma_{A_{i_t}}(v_{i_t}, a_t)$ for $t \in [c-1]$ and $v' \in \Gamma_{A_{i_c}}(v_{i_c}, a_c)$. Consider an index $t \in [c-1]$. Since $(i_1, \dots, i_c)$ is a good sequence, any vertex u on π with $v_{i_t} \prec_\pi u \prec_\pi v_{i_{t+1}}$ is contained in A_{i_t}. Equivalently, any internal vertex of π_t is contained in A_{i_t}. Thus, π_t witnesses that $v_{i_{t+1}} \in \Gamma_{A_{i_t}}(v_{i_t}, a_t)$. Furthermore, we observe that all internal vertices of π_c are contained in A_{i_c}. To see this, assume some internal vertex of π_c is not in A_{i_c}. Let u' be the first internal vertex on π_c that is not in A_{i_c} (i.e., the one closest to the v_{i_c}-end of π_c). Since $u' \in A_i$, we have $u' = v_j$ for some $j \in [i-1]$. Also, since $u' \notin A_{i_c}$, we have $j \geq i_c$, which implies $j > i_c$ as $j \neq i_c$. By the choice of u', it holds that $u \in A_{i_c}$ for all u satisfying $v_{i_c} \prec_\pi u \prec_\pi u' = v_j$. Therefore, $(i_1, \dots, i_c, j)$ is a good sequence of length $c+1$, which contradicts the definition of c. As such, we see that all internal vertices of π_c are contained in A_{i_c}. This further implies that $v' \in \Gamma_{A_{i_c}}(v_{i_c}, a_c)$ and hence $(v_{i_1}, \dots, v_{i_c}, v')$ is an L-type sequence.

By the above argument, we can now charge each vertex $v' \in \Gamma_{A_i}(v,\ell)\backslash \mathrm{SR}_\ell(G,\sigma,v)$ to some $L \in \mathcal{L}_{<\ell}$ such that there exists an L-type sequence ending at v'. Note that each $L \in \mathcal{L}_{<\ell}$ gets charged at most $\#_L$ times, since the number of different L-type sequences is $\#_L$. Recall that $\#_L \leq \Pi_L^{(p)} \cdot (p-1)$. Therefore, we have

$$|\Gamma_{A_i}(v,\ell)\backslash \mathrm{SR}_\ell(G,\sigma,v)| \leq \sum_{L \in \mathcal{L}_{<\ell}} \#_L \leq \sum_{L \in \mathcal{L}_{<\ell}} \Pi_L^{(p)} \cdot (p-1).$$

Furthermore, $|\mathrm{SR}_\ell(G,\sigma,v)| \leq \mathrm{scol}_\ell(G,\sigma) \leq \mathrm{scol}_k(G,\sigma) = p$. It follows that

$$|\Gamma_{A_i}(v,\ell)| \leq p + \sum_{L \in \mathcal{L}_{<\ell}} \Pi_L^{(p)} \cdot (p-1).$$

By Fact 5, the above inequality implies $|\Gamma_{A_i}(v,\ell)| \leq C_\ell^{(p)}$. As $|\Gamma_{A_i}(v,\ell)| \leq C_\ell^{(p)}$ for all $\ell \in [k]$, we can successfully construct v_i in the i-th iteration of the for-loop. As a result, Algorithm 1 terminates with $p \leq \mathrm{scol}_k(G)$. □

Suppose Algorithm 1 terminates with $p \leq \mathrm{scol}_k(G)$. Let $\sigma = (v_1,\dots,v_n)$ be the ordering of $V(G)$ returned by algorithm, and define $A_i = \{v_1,\dots,v_{i-1}\}$ for $i \in [n]$. By the construction procedure of the algorithm, we have $|\Gamma_{A_i}(v_i,\ell)| \leq C_\ell^{(p)}$ for all $\ell \in [k]$. Note that $\Gamma_{A_i}(v_i,\ell) = \mathrm{SR}_\ell(G,\sigma,v_i)$ for $i \in [n]$ and $\ell \in [k]$. Thus, $\mathrm{scol}_\ell(G,\sigma) \leq C_\ell^{(p)}$ for all $\ell \in [k]$, and in particular,

$$\mathrm{scol}_k(G,\sigma) \leq C_k^{(p)} = (2p)^{O(k)}.$$

Since $p \leq \mathrm{scol}_k(G)$ and we may assume $\mathrm{scol}_k(G) \geq 2$ (for otherwise G has no edge and the problem is trivial), we have $\mathrm{scol}_k(G,\sigma) = \mathrm{scol}_k^{O(k)}(G)$, which proves the correctness of our algorithm.

We omit the linear-time implementation of our algorithm is it is somehow complicated.

2.2 New approach for subgraph isomorphism

Let H be a pattern graph with k vertices. For a subset $V \subseteq V(H)$, denote by $N_H[V] \subseteq V(H)$ the closed neighborhood of V in H, which consists of the vertices that are either in V or neighboring to some vertex in V. For each ordering σ of $V(H)$, we define a corresponding tree decomposition $\mathcal{T}_H^\sigma$ of H, which we call the *canonical tree decomposition* of H under σ. Suppose $\sigma = (v_1,\dots,v_k)$. A *splitter* of H is a subset $X \subseteq V(H)$ such that $N_H[V(C)] \subsetneq V(H)$ for every connected component C of $H - X$. Note that if $H - X$ contains more than one connected components, then X is necessarily a splitter of H. Define $c \in [k]_0$ as the largest number such that $\{v_{c+1},\dots,v_k\}$ is a splitter of H, and let $X = \{v_{c+1},\dots,v_k\}$. Such a number c must exist, since $\{v_1,\dots,v_k\}$ is trivially a splitter of H by definition. We call X the *canonical splitter* of H under σ.

Lemma 7. *$X \subseteq SR_k(H,\sigma,v_{c+1})$.*

Proof. Consider a vertex $v_i \in X$. We have $i \geq c+1$. If $i = c+1$, then $v_i = v_{c+1} \in \mathrm{SR}_k(H,\sigma,v_{c+1})$. So assume $i > c+1$. Observe that $X\backslash\{v_{c+1}\} = \{v_{c+2},\dots,v_k\}$ is not a splitter of H, because of the maximality of c. This implies that the graph $H' = H - (X\backslash\{v_{c+1}\})$ is connected and $N_H[V(H')] = N_H[\{v_1,\dots,v_{c+1}\}] = V(H)$. So for any vertex $v \in V(H)$, there is a path between v_{c+1} and v whose internal vertices are in H'. In particular, we have a path π between v_{c+1} and v_i with internal vertices belonging to H'. Without loss of generality, we may assume π is simple, and thus the length of π is at most k. Now v_i is larger than v_{c+1} under σ, and the internal vertices of π are all smaller than v_{c+1} under σ, because v_{c+1} is the largest vertex in H'. Thus, the path π witnesses the fact that $v_i \in \mathrm{SR}_k(H,\sigma,v_{c+1})$. □

Now we construct the canonical tree decomposition $\mathcal{T}_H^\sigma$. We create a root node t with bag $\beta(t) = X$. Suppose $H-X$ has connected components $C_1,\dots,C_p$. Let H_i be the induced subgraph of H with vertex set $N_H[V(C_i)]$, and σ_i be the ordering of $V(H_i)$ inherited from σ. As X is a splitter of H, we have $|V(H_i)| \leq k-1$ for all $i \in [p]$. We recursively construct the canonical tree decomposition of H_i under σ_i, $\mathcal{T}_{H_i}^{\sigma_i}$. Then we append $\mathcal{T}_{H_1}^{\sigma_1},\dots,\mathcal{T}_{H_p}^{\sigma_p}$ as the subtrees of the root node t. This completes the construction of $\mathcal{T}_H^\sigma$.

It is not difficult to prove that $\mathcal{T}_H^\sigma$ is a tree decomposition of H. The only property that is a bit non-obvious is that the nodes whose bags contain a vertex $v \in V(H)$ form a connected subtree. We briefly discuss why this is the case. Because of our recursive construction, by induction, we actually only need to verify the property for vertices $v \in X$. We claim that if $v \in V(H_i)$, then v is contained in the root bag of $\mathcal{T}_{H_i}^{\sigma_i}$. Recall that $V(H_i) = N_H[V(C_i)]$. Let $Y \subseteq V(H_i)$ be the canonical splitter of H_i under σ_i, which is the root bag of $\mathcal{T}_{H_i}^{\sigma_i}$. By the construction of canonical splitters, Y contains the largest vertices of H_i under σ. Thus, to see $v \in Y$, it suffices to show that Y contains some vertex of H_i smaller than v. As $v \in X$, all vertices in C_i are smaller than v. Furthermore, observe that Y must contain some vertex in C_i. Otherwise, Y cannot be a splitter of H_i, because $N_{H_i}[V(C_i)] = N_H[V(C_i)] = V(H_i)$. Therefore, Y contains some vertex smaller than v and thus contains v. Now we see that if v is contained in some bag of $\mathcal{T}_{H_i}^{\sigma_i}$, then it must be contained in the root bag of $\mathcal{T}_{H_i}^{\sigma_i}$ as well. By induction, we can assume the nodes of $\mathcal{T}_{H_i}^{\sigma_i}$ whose bags contain v form a connected subtree for each $i \in [p]$. It then follows that the nodes of $\mathcal{T}_H^\sigma$ whose bags contain v form a connected subtree.

The canonical tree decomposition $\mathcal{T}_H^\sigma$ has several nice properties, among which the most important one is the following: the vertices in every bag are strongly reachable from the smallest vertex in the bag (under σ) in k steps. Formally, suppose $\mathcal{T}_H^\sigma = (T,\beta)$ where T is the tree and $\beta : T \to 2^{V(H)}$ defines the bags. Then we have $\beta(t) \subseteq \mathrm{SR}_k(H,\sigma,\min_\sigma(\beta(t)))$ for all $t \in T$, where $\min_\sigma(\beta(t))$ denotes the smallest vertex in $\beta(t)$ under σ. This property follows immediately from Lemma 7. There are also some other properties needed for our algorithms, but we omit them in this overview for simplicity.

Next, we discuss how to use canonical tree decompositions to solve Subgraph Isomorphism. Let G be the host graph with n vertices with an ordering σ of $V(G)$. We describe an algorithm that finds a subgraph isomorphism from H to G whose time cost depends on $\mathrm{scol}_k(G,\sigma)$. Each injective map $\pi : V(H) \to V(G)$ transfers the ordering σ of $V(G)$ to an ordering of $V(H)$, which we denote by $\pi^{-1}(\sigma)$. For a tree decomposition $\mathcal{T}$ of H, we say a subgraph isomorphism $\pi : V(H) \to V(G)$ from H to G is of $\mathcal{T}$-type if $\mathcal{T}_H^{\pi^{-1}(\sigma)} = \mathcal{T}$. It suffices to consider every tree decomposition $\mathcal{T}$ of H, and find a subgraph isomorphism from H to G of $\mathcal{T}$-type. (For counting, we just sum up the number of subgraph isomorphisms from H to G of $\mathcal{T}$-type for all tree decompositions $\mathcal{T}$ of H.)

Fix a tree decomposition $\mathcal{T} = (T, \beta)$ of H, and assume that there exists at least one subgraph isomorphism from H to G that is of $\mathcal{T}$-type. For a node $t \in T$, let T_t be the subtree of T rooted at t and $V_t = \bigcup_{t' \in T_t} \beta(t')$. We now sketch a simple algorithm that takes $G, H, \mathcal{T}$ as input and finds a subgraph isomorphism from H to G, which is *not* necessarily of $\mathcal{T}$-type. (In our actual algorithm, we can guarantee the subgraph isomorphism found to be of $\mathcal{T}$-type, but it requires additional properties of canonical tree decompositions. This feature is not important for finding subgraph isomorphisms, but it becomes crucial when solving the counting version.) Consider a (unknown) subgraph isomorphism $\pi : V(H) \to V(G)$ from H to G of $\mathcal{T}$-type. We know that $\mathcal{T}_H^{\pi^{-1}(\sigma)} = \mathcal{T}$. By the nice property of $\mathcal{T}_H^{\pi^{-1}(\sigma)}$, we have $\beta(t) \subseteq \mathrm{SR}_k(H, \pi^{-1}(\sigma), \min_{\pi^{-1}(\sigma)}(\beta(t)))$ for all $t \in T$. As π is a subgraph isomorphism, this implies $\pi(\beta(t)) \subseteq \mathrm{SR}_k(G, \sigma, \min_\sigma(\pi(\beta(t))))$ for all $t \in T$. Based on this, we solve the problem as follows. For convenience, write $S_v = \mathrm{SR}_k(G, \sigma, v)$ for $v \in V(G)$. Note that $|S_v| \leq \mathrm{scol}_k(G, \sigma)$. We say a map $f : V \to V(G)$ where $V \subseteq V(H)$ is *well-behaved* if for every $t \in T$ such that $V \cap \beta(t) \neq \emptyset$, we have $f(V \cap \beta(t)) \subseteq S_v$ where $v = \min_\sigma(f(V \cap \beta(t)))$. According to the discussion above, we see there exists at least one well-behaved subgraph isomorphism from H to G. For each node $t \in T$ and each well-behaved subgraph isomorphism $\phi : \beta(t) \to V(G)$ from $H[\beta(t)]$ to G, define $\mathcal{A}_{t,\phi}$ as the collection of all well-behaved subgraph isomorphism $\pi : V_t \to V(G)$ from $H[V_t]$ to G such that $\phi = \pi_{|\beta(t)}$. We say a pair (t, ϕ) is *relevant* if $\mathcal{A}_{t,\phi}$ is defined. For each $t \in T$, the number of well-behaved subgraph isomorphisms $\phi : \beta(t) \to V(G)$ from $H[\beta(t)]$ to G is bounded by $\sum_{v \in V(G)} |S_v|^k \leq \mathrm{scol}_k^k(G, \sigma) \cdot n$. The number of nodes in T is, in fact, $O(k)$ in our construction of canonical tree decomposition. Therefore, the total number of relevant pairs is at most $O(\mathrm{scol}_k^k(G, \sigma) \cdot kn)$, so is the number of collections $\mathcal{A}_{t,\phi}$. Our algorithm processes the nodes of T in a bottom-up order. When processing a node $t \in T$, we consider every relevant pairs (t, ϕ), and try to compute a $(k - |V_t|)$-representative set $\mathcal{A}'_{t,\phi} \subseteq \mathcal{A}_{t,\phi}$. (When defining representative sets we view $\mathcal{A}_{t,\phi}$ as a collection of size-$|V_t|$ subsets of $V(G)$ which are the images of the maps in $\mathcal{A}_{t,\phi}$.) Note that if s is a child of t in T, then for each $\pi \in \mathcal{A}_{t,\phi}$, we have $\pi_{|V_s} \in \mathcal{A}_{s,\tau}$ where $\tau = \pi_{|\beta(s)}$. In other words, the maps in $\mathcal{A}_{t,\phi}$ can be obtained by "gluing" the maps in $\mathcal{A}_{s,\tau}$ for children s of t together with ϕ. As such, the properties of representative sets allow us to efficiently compute $\mathcal{A}'_{t,\phi}$ by "gluing" the representative sets $\mathcal{A}'_{s,\tau}$ for children s of t (which we have already computed), and we can guarantee that $|\mathcal{A}'_{t,\phi}| \leq \binom{k}{|V_t|} \leq 2^k$. We omit the details here as this step is somehow standard. With a careful implementation, considering each relevant pair only takes $2^{O(k)}$ time. So the time cost for computing all $\mathcal{A}'_{t,\phi}$ is $\mathrm{scol}_k^{O(k)}(G, \sigma) \cdot n$. After this, we consider the root t of T, find a relevant pair (t, ϕ) such that $\mathcal{A}'_{t,\phi} \neq \emptyset$, and return an element in $\mathcal{A}'_{t,\phi}$ (which is clearly a subgraph isomorphism from $H[V_t] = H$ to G). We know there exists at least one well-behaved subgraph isomorphism $\pi : V(H) \to V(G)$ from H to G. Thus, $\mathcal{A}_{t,\phi} \neq \emptyset$ where t is the root of T and $\phi = \pi_{|\beta(t)}$. As $\mathcal{A}'_{t,\phi}$ is a 0-representative set of $\mathcal{A}_{t,\phi}$, it must be nonempty. So our algorithm is guaranteed to find a solution.

The above algorithm works with a specific tree decomposition $\mathcal{T}$ of H, and we still need to guess $\mathcal{T}$. However, we do not need to try every tree decomposition of H. Indeed, if a subgraph isomorphism from H to G is of $\mathcal{T}$-type, then we must have $\mathcal{T} \in \mathbf{T}_H = \{\mathcal{T}_H^\eta : \eta \text{ is an ordering of } V(H)\}$. Therefore, trying the tree decompositions in $\mathbf{T}_H$ suffices. As $|\mathbf{T}_H| \leq k!$, this gives us an extra $k^{O(k)}$ factor in the running time. So the overall time complexity of our algorithm is $(k + \mathrm{scol}_k(G, \sigma))^{O(k)} \cdot n$. Using our ordering algorithm, we can compute in $\mathrm{scol}_k^{O(k)}(G) \cdot n$ time an ordering σ of $V(G)$ with $\mathrm{scol}_k(G, \sigma) = \mathrm{scol}_k(G)$, resulting in a running time of $(k + \mathrm{scol}_k(G))^{O(k^2)} \cdot n$.

2.3 Distance-based representative sets/functions

We give an overview of how to construct (a simplified version of) distance-based representative sets/functions in sparse graphs. We mainly focus on representative sets, and our approach directly works for representative functions as well. Let G be a graph. Recall the following definitions. Two subsets $P, Q \subseteq V(G)$ are *r-far* in G if $\mathrm{dist}_G(u, v) \geq r$ for all $u \in P$ and $v \in Q$. For a collection $\mathcal{P}$ of subsets of $V(G)$, a distance-r q-representative set of $\mathcal{P}$ is a sub-collection $\mathcal{P}' \subseteq \mathcal{P}$ satisfying the following: for any $Q \subseteq V(G)$ with $|Q| = q$, if there exists $P \in \mathcal{P}$ such that P and Q are r-far in G, then there also exists $P' \in \mathcal{P}'$ such that P' and Q are r-far in G. As in the standard representative sets, we are interested in the case where all sets in $\mathcal{P}$ are small, say, of size p.

We first observe that if G is a general graph, then one cannot hope any small-size distance-r q-representative sets even for $p = q = 1$ and $r = 2$. (Note that when $r = 1$, distance-r q-representative sets are just the standard q-representative sets and one can construct such representative sets of size $\binom{p+q}{p}$ [15].) Let G be a graph with $2n$ vertices $u_1, \dots, u_n, v_1, \dots, v_n$ and edges (u_i, v_j) for all $i, j \in [n]$ with $i \neq j$. In other words, G is obtained from a biclique $K_{n,n}$ by removing a perfect matching. Define $\mathcal{P} = \{\{u_i\} : i \in [n]\}$. Observe that any distance-2 1-representative set $\mathcal{P}' \subseteq \mathcal{P}$ must contain all sets in $\mathcal{P}$. To see this, assume $\{u_i\} \notin \mathcal{P}'$. Let $Q = \{v_i\}$. Then we have $P = \{u_i\} \in \mathcal{P}$ such that P and Q are 2-far in G, but we do not have such a set in $\mathcal{P}'$.

The requirement for G we need is that the $(r-1)$-th weak coloring number of G, $\mathrm{wcol}_{r-1}(G)$, is a constant (which is the case if G is taken from a graph class of bounded expansion). In other words, the size of our distance-based representative sets relies on $\mathrm{wcol}_{r-1}(G)$. The way of defining weak coloring numbers is similar to strong coloring numbers. Let σ be an ordering of $V(G)$. For $u, v \in V(G)$, we say u is *weakly reachable* from v within ℓ steps under σ if $u \geq_\sigma v$ and there exists a path in G between u and v of length at most ℓ whose internal vertices are all smaller than u under σ. Denote by $\mathrm{WR}_\ell(G, \sigma, v)$ the set of all vertices of G weakly reachable from v within ℓ steps under σ. Define $\mathrm{wcol}_\ell(G, \sigma) = \max_{v \in V(G)} |\mathrm{WR}_\ell(G, \sigma, v)|$. The ℓ-th *weak coloring number* of G, denoted by $\mathrm{wcol}_\ell(G)$, is the minimum of $\mathrm{wcol}_\ell(G, \sigma)$ over all orderings σ of $V(G)$.

Suppose $\mathrm{wcol}_{r-1}(G) = \Delta$. Fix an ordering σ of G such that $\mathrm{wcol}_{r-1}(G, \sigma) = \mathrm{wcol}_{r-1}(G) = \Delta$. Let $\mathcal{P}$ be a collection of subsets of $V(G)$ each of which is of size p. To construct a small distance-r q-representative set of $\mathcal{P}$, the key idea is to reduce to *standard* representative sets. Write $U_G = V(G) \times [r]$. For each $P \in \mathcal{P}$, we define a set

$$\alpha(P) = \{(v, c) \in U_G : \text{there is } x \in P \text{ with } v \in \mathrm{WR}_{c-1}(G, \sigma, x)\},$$

which is a subset of U_G. On the other hand, for each $Q \subseteq V(G)$, we define a set

$$\beta(Q) = \{(v, c) \in U_G : \text{there is } y \in Q \text{ with } v \in \mathrm{WR}_{r-c}(G, \sigma, y)\}.$$

We have the following key observations.

Lemma 8. *P and Q are r-far in G if and only if $\alpha(P) \cap \beta(Q) = \emptyset$.*

Proof. To see the "if" direction, suppose $\alpha(P) \cap \beta(Q) = \emptyset$. Let $x \in P$ and $y \in Q$. We want to show $\mathrm{dist}_G(x, y) \geq r$. Assume there exists a path π between x and y in G of length $\ell < r$. Without loss of generality, assume π is simple. Let v be the largest vertex on π under σ. Then v subdivides π into two subpaths π_1 and π_2, where π_1 (resp., π_2) connects x and v (resp., y and v). Suppose π_1 is of length $c - 1$. Then π_2 is of length $\ell - c + 1$. The largest vertex on π_1 under σ is v, which implies $v \in \mathrm{WR}_{c-1}(G, \sigma, x)$. Thus, $(v, c) \in \alpha(P)$. On the other hand, the largest vertex on π_2 under σ is also v, which implies $v \in \mathrm{WR}_{\ell-c+1}(G, \sigma, y)$. As $\ell < r$, we have $\mathrm{WR}_{\ell-c+1}(G, \sigma, y) \subseteq \mathrm{WR}_{r-c}(G, \sigma, y)$ and hence $v \in \mathrm{WR}_{r-c}(G, \sigma, y)$. Thus, $(v, c) \in \beta(Q)$. This contradicts the fact $\alpha(P) \cap \beta(Q) = \emptyset$.

To see the "only if" direction, suppose P and Q are r-far in G. Assume $\alpha(P) \cap \beta(Q) \neq \emptyset$ and let $(v, c) \in \alpha(P) \cap \beta(Q)$. As $(v, c) \in \alpha(P)$, there exists $x \in P$ such that $\mathrm{dist}_G(x, v) \leq c - 1$. Similarly, as $(v, c) \in \beta(Q)$, there exists $y \in P$ such that $\mathrm{dist}_G(v, y) \leq r - c$. Therefore, $\mathrm{dist}_G(x, y) \leq \mathrm{dist}_G(x, v) + \mathrm{dist}_G(v, y) \leq r - 1$. This contradicts our assumption that P and Q are r-far in G. □

Lemma 9. *$|\alpha(P)| \leq r\Delta|P|$ and $|\beta(Q)| \leq r\Delta|Q|$.*

Proof. If $(v, c) \in \alpha(P)$, then $v \in \mathrm{WR}_{r-1}(G, \sigma, x)$ for some $x \in P$. So for a fixed $c \in [r]$, there can be at most $\sum_{x \in P} |\mathrm{WR}_{r-1}(G, \sigma, x)| \leq \Delta|P|$ vertices $v \in V(G)$ such that $(v, c) \in \alpha(P)$. This implies $|\alpha(P)| \leq r\Delta|P|$. Similarly, if $(v, c) \in \beta(Q)$, then $v \in \mathrm{WR}_{r-1}(G, \sigma, y)$ for some $y \in Q$. So for a fixed $c \in [r]$, there can be at most $\sum_{y \in Q} |\mathrm{WR}_{r-1}(G, \sigma, y)| \leq \Delta|Q|$ vertices $v \in V(G)$ such that $(v, c) \in \beta(Q)$. This implies $|\beta(Q)| \leq r\Delta|Q|$. □

Based on the above lemmas, it is easy to see that a distance-r q-representative set of $\mathcal{P}$ can be obtained from a standard $(r\Delta q)$-representative set of $\alpha(\mathcal{P}) = \{\alpha(P) : P \in \mathcal{P}\}$. Specifically, consider a sub-collection $\mathcal{P}' \subseteq \mathcal{P}$ such that $\alpha(\mathcal{P}') \subseteq \alpha(\mathcal{P})$ is a $(r\Delta q)$-representative set of $\alpha(\mathcal{P})$. We claim that $\mathcal{P}'$ is a distance-r q-representative set of $\mathcal{P}$. To see this, let $Q \subseteq V(G)$ with $|Q| = q$. Suppose there exists $P \in \mathcal{P}$ such that P and Q are r-far in G. By Lemma 8, $\alpha(P) \cap \beta(Q) = \emptyset$. As $\alpha(\mathcal{P}')$ is a $(r\Delta q)$-representative set of $\alpha(\mathcal{P})$ and $|\beta(Q)| \leq r\Delta q$ by Lemma 9, there should exist $P' \in \mathcal{P}'$ such that $\alpha(P') \cap \beta(Q) = \emptyset$. Using Lemma 8 again, we see P' and Q are r-far in G. Thus, $\mathcal{P}'$ is a distance-r q-representative set of $\mathcal{P}$. Note that all sets in $\alpha(\mathcal{P})$ are of size at most $r\Delta p$ by Lemma 9. Therefore, there exists a $(r\Delta q)$-representative set of $\alpha(\mathcal{P})$ with size $\binom{r\Delta p + r\Delta q}{r\Delta p} = 2^{O(p+q)}$ which can be computed efficiently [15], assuming r and Δ are constants. Then mapping such a representative set back to a sub-collection of $\mathcal{P}$ gives us a distance-r q-representative set of $\mathcal{P}$ with the same size. This completes our construction of distance-based representative sets.

As our approach "transfers" the r-farness among subsets of $V(G)$ to disjointness of subsets of U_G, it can be directly extended to give distance-based representative functions as well, by using the known constructions for (standard) representative functions [21, 22].

Finally, we notice that when $r = 2$, our construction only requires $\mathsf{wcol}_{r-1}(G) = \mathsf{wcol}_1(G)$ to be bounded, and $\mathsf{wcol}_1(G)$ is nothing but the *degeneracy* of G. Therefore, our construction can give small-size distance-2 representative sets/functions for degenerate graphs. This serves an important role when proving Theorem 3.

3 CONCLUSION

This paper studied a broad family of first-order model checking problems, Dist-r Pattern Finding, and their counting versions, Dist-r Pattern Counting, on sparse graphs. This subsumes all problems that can be formulated as finding/counting k vertices in a host graph with certain distance constraints. Our main result is a new algorithmic framework for Dist-r Pattern Finding/Counting that gives algorithms with running time $2^{k^{O(1)}} n$ on all exponential-expansion graph classes. In terms of techniques, our framework differs entirely from the ones based on centered coloring. Instead, we developed several new tools, including a new vertex-ordering algorithm that approximates the generalized coloring numbers, distance-based representative sets/functions, etc. We believe that these tools are of independent interests and can find applications in other related problems.

ACKNOWLEDGMENTS

The research of Saket Saurabh was partially supported by the European Research Council (ERC) grant no. 819416, and Swarnajayanti Fellowship no. DST/SJF/MSA01/2017-18. The research of Meirav Zehavi was partially supported by the European Research Council (ERC) grant no. 101039913. The authors would like to thank the anonymous reviewers for their insightful comments on this paper.

REFERENCES

[1] Suman K Bera, Lior Gishboliner, Yevgeny Levanzov, C Seshadhri, and Asaf Shapira. 2022. Counting subgraphs in degenerate graphs. *ACM Journal of the ACM (JACM)* 69, 3 (2022), 1–21.

[2] Marco Bressan. 2021. Faster algorithms for counting subgraphs in sparse graphs. *Algorithmica* 83 (2021), 2578–2605.

[3] M Bressan, LA Goldberg, K Meeks, and M Roth. 2024. Counting subgraphs in somewhere dense graphs. *SIAM J. Comput.* (2024).

[4] Marco Bressan and Marc Roth. 2022. Exact and approximate pattern counting in degenerate graphs: New algorithms, hardness results, and complexity dichotomies. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 276–285.

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

[6] Zdenek Dvorak. 2011. Constant-factor approximation of domination number in sparse graphs. arXiv:1110.5190 [math.CO]

[7] Zdeněk Dvořák. 2013. Constant-factor approximation of the domination number in sparse graphs. *European Journal of Combinatorics* 34, 5 (2013), 833–840.

[8] Zdeněk Dvořák. 2021. Approximation metatheorems for classes with bounded expansion. *arXiv preprint arXiv:2103.08698* (2021).

[9] Zdeněk Dvořák, Daniel Král, and Robin Thomas. 2013. Testing first-order properties for subclasses of sparse graphs. *Journal of the ACM (JACM)* 60, 5 (2013), 1–24.

[10] Zdeněk Dvořák and Abhiruk Lahiri. 2021. Approximation schemes for bounded distance problems on fractionally treewidth-fragile graphs. *arXiv preprint arXiv:2105.01780* (2021).

[11] Zdeněk Dvořák, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. 2023. Efficient Approximation for Subgraph-Hitting Problems in Sparse Graphs and Geometric Intersection Graphs. *arXiv preprint arXiv:2304.13695* (2023).

[12] Zdeněk Dvořák and Sergey Norin. 2016. Strongly Sublinear Separators and Polynomial Expansion. *SIAM Journal on Discrete Mathematics* 30, 2 (2016), 1095–1101.

[13] Louis Esperet and Jean-Florent Raymond. 2018. Polynomial expansion and sublinear separators. *European Journal of Combinatorics* 69 (2018), 49–53.

[14] Fedor V Fomin, Alexander Golovnev, Alexander S Kulikov, and Ivan Mihajlin. 2015. Tight bounds for subgraph isomorphism and graph homomorphism. *arXiv preprint arXiv:1507.03738* (2015).

[15] Fedor V Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. 2016. Efficient computation of representative families with applications in parameterized and exact algorithms. *Journal of the ACM (JACM)* 63, 4 (2016), 1–60.

[16] Martin Grohe, Stephan Kreutzer, Roman Rabinovich, Sebastian Siebertz, and Konstantinos Stavropoulos. 2016. Colouring and covering nowhere dense graphs. In *Graph-Theoretic Concepts in Computer Science: 41st International Workshop, WG 2015, Garching, Germany, June 17-19, 2015, Revised Papers 41*. Springer, 325–338.

[17] Martin Grohe, Stephan Kreutzer, Roman Rabinovich, Sebastian Siebertz, and Konstantinos Stavropoulos. 2018. Coloring and covering nowhere dense graphs. *SIAM Journal on Discrete Mathematics* 32, 4 (2018), 2467–2481.

[18] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. 2017. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM (JACM)* 64, 3 (2017), 1–32.

[19] Sariel Har-Peled and Kent Quanrud. 2017. Approximation algorithms for polynomial-expansion and low-density graphs. *SIAM J. Comput.* 46, 6 (2017), 1712–1744.

[20] Hal A Kierstead and Daqing Yang. 2003. Orderings on graphs and game coloring number. *Order* 20 (2003), 255–264.

[21] Daniel Lokshtanov, Andreas Björklund, Saket Saurabh, and Meirav Zehavi. 2021. Approximate Counting of k-Paths: Simpler, Deterministic, and in Polynomial Space. *ACM Trans. Algorithms* 17, 3 (2021), 26:1–26:44.

[22] Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. 2021. Efficient computation of representative weight functions with applications to parameterized counting (extended version). In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 179–198.

[23] Gary L Miller, Shang-Hua Teng, William Thurston, and Stephen A Vavasis. 1997. Separators for sphere-packings and nearest neighbor graphs. *Journal of the ACM (JACM)* 44, 1 (1997), 1–29.

[24] Jesper Nederlof. 2020. Detecting and counting small patterns in planar graphs in subexponential parameterized time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 1293–1306.

[25] Jaroslav Nešetřil and Patrice Ossona De Mendez. 2008. Grad and classes with bounded expansion I. Decompositions. *European Journal of Combinatorics* 29, 3 (2008), 760–776.

[26] Jaroslav Nešetřil and Patrice Ossona De Mendez. 2008. Grad and classes with bounded expansion II. Algorithmic aspects. *European Journal of Combinatorics* 29, 3 (2008), 777–791.

[27] Jaroslav Nešetřil and Patrice Ossona De Mendez. 2011. On nowhere dense graphs. *European Journal of Combinatorics* 32, 4 (2011), 600–617.

[28] Jaroslav Nešetřil and Patrice Ossona De Mendez. 2012. *Sparsity: graphs, structures, and algorithms*. Vol. 28. Springer Science & Business Media.

[29] Jaroslav Nešetřil, Patrice Ossona de Mendez, and David R Wood. 2012. Characterisations and examples of graph classes with bounded expansion. *European Journal of Combinatorics* 33, 3 (2012), 350–373.

[30] Felix Reidl and Blair D Sullivan. 2023. A color-avoiding approach to subgraph counting in bounded expansion classes. *Algorithmica* 85, 8 (2023), 2318–2347.

[31] Felix Reidl, Fernando Sánchez Villaamil, and Konstantinos Stavropoulos. 2019. Characterising bounded expansion by neighbourhood complexity. *European Journal of Combinatorics* 75 (2019), 152–168.

[32] Xuding Zhu. 2009. Colouring graphs with bounded generalized colouring number. *Discrete Mathematics* 309, 18 (2009), 5562–5568.

Received 2024-11-04; accepted 2025-02-01