

Randomized Communication and Implicit Graph Representations*

Nathaniel Harms[†]
EPFL, Switzerland
nathaniel.harms@epfl.ch

Sebastian Wild[‡]
University of Marburg, Germany &
University of Liverpool, UK
wild@informatik.uni-marburg.de

Viktor Zamaraev
University of Liverpool, UK
viktor.zamaraev@liverpool.ac.uk

December 12, 2024

Abstract

We initiate the focused study of constant-cost randomized communication, with emphasis on its connection to graph representations. We observe that constant-cost randomized communication problems are equivalent to hereditary (i.e. closed under taking induced subgraphs) graph classes which admit constant-size *adjacency sketches* and *probabilistic universal graphs* (PUGs), which are randomized versions of the well-studied adjacency labeling schemes and induced-universal graphs. This gives a new perspective on long-standing questions about the existence of these objects, including new methods of constructing adjacency labeling schemes.

We ask three main questions about constant-cost communication, or equivalently, constant-size PUGs: (1) Are there any natural, non-trivial problems aside from Equality and k -Hamming Distance which have constant-cost communication? We provide a number of new examples, including deciding whether two vertices have path-distance at most k in a planar graph, and showing that constant-size PUGs are preserved by the Cartesian product operation. (2) What structures of a problem explain the existence or non-existence of a constant-cost protocol? We show that in many cases a Greater-Than subproblem is such a structure. (3) Is the Equality problem *complete* for constant-cost randomized communication? We show that it is not: there are constant-cost problems which do not reduce to Equality.

*An earlier version of this paper appeared at STOC 2022.

[†]This work was done while the author was a student at the University of Waterloo. Partly supported by an NSERC postgraduate scholarship.

[‡]The second author is supported by EPSRC grant EP/X039447/1.

Contents

1	Introduction	1
1.1	Communication Complexity	2
1.2	Implicit Graph Representations	5
1.3	Subsequent Work	7
1.4	Organization	8
2	Preliminaries	9
2.1	Notation and Terminology	9
2.2	Communication Complexity	10
2.3	Adjacency Labeling Schemes, Universal Graphs, and Factorial Classes	10
3	Connections: Randomized Communication and Implicit Representations	11
3.1	Adjacency Sketching and Probabilistic Universal Graphs	12
3.2	Communication-to-Graph Correspondence	13
3.3	Reductions	16
3.4	Equality-Based Labeling Schemes	19
3.5	Basic Adjacency Sketches	20
4	Question I. New Examples of Constant-Cost Communication	21
4.1	Computing Small Distances and First-Order Formulae	21
4.2	Graph Products	28
5	Question II. Structure: Stability is Sometimes Sufficient	32
5.1	Interval & Permutation Graphs	33
5.2	Monogenic Bipartite Graphs	42
6	Question III. Equality is Not Complete	55
A	Missing Proofs from Section 3	62
A.1	Probability Boosting and Derandomization	62
A.2	Lower Bound from GREATER-THAN	63
A.3	Missing Proof for Equality-Based Labeling	65
B	The Lattice of Hereditary Graph Classes	65
B.1	The Speed of Hereditary Graph Families	65
B.2	Minimal Factorial Families	67

1 Introduction

Randomized communication is a central topic in communication complexity [GPW18, CLV19, PSW20, HHH23], and yet the power of randomness in communication remains poorly understood. The textbook example of the power of randomness in communication is the EQUALITY problem (where two players wish to decide if they have received identical inputs), which has a *constant-cost* randomized protocol (see e.g. [NK96]): the required number of bits of communication remains constant, regardless of input size, whereas a deterministic communication protocol cannot do better than simply sending the entire input.

To better understand the power of randomness in communication, we wish to understand the most extreme examples. Therefore the goal of this paper is to initiate the focused study of the communication problems that, like EQUALITY, have constant-cost protocols. In related models of computation like randomized decision trees or even *deterministic* communication, the class of constant-cost problems is trivial¹. *A priori*, one may also expect the class of constant-cost randomized communication problems to be simple or nearly trivial, but we will see that this is not the case: the class is surprisingly rich, containing a variety of natural problems and having many connections to other areas.

One of the central observations of this paper is that constant-cost randomized communication problems are equivalent to hereditary graph classes which admit a constant-size *adjacency sketch*, a probabilistic version of an *adjacency labeling scheme* (also known as an *implicit representation*), the subject of a large body of research in graph theory and distributed computing (e.g. [KNR92, FK09, KM12, ACLZ15, Har20, BGK⁺21, DEG⁺21, HH22]). Therefore the study of constant-cost randomized communication can be viewed from two perspectives, the communication complexity perspective and the graph representation perspective. In Section 1.1 we describe our motivations and results from the perspective of communication complexity, and in Section 1.2 from the perspective of graph representations. Each of these perspectives suggest different questions, but, to begin understanding constant-cost randomized communication, we believe it is important to answer the following 3 questions, which will be the focus of this paper:

Question I. Are there natural examples of constant-cost randomized communication problems, that are not trivially obtained from the few known examples (EQUALITY and k -HAMMING DISTANCE)? Equivalently, what hereditary classes of graphs admit constant-size adjacency sketches? We will give a number of new examples, which also lead to new techniques for obtaining implicit representations of graphs.

Question II. What structures of problems determine the existence or non-existence of constant-cost randomized communication? One of the main applications of communication complexity is to prove lower bounds, and constant vs. non-constant is the most basic lower-bound question. One might therefore hope for a simple criterion to determine whether a problem has constant cost or not.

Question III. What “types” of constant-cost randomized protocols are there? In particular, is there a “complete” problem for this class, which all constant-cost problems reduce to? The most natural candidate is the EQUALITY problem, and we show that EQUALITY is *not* complete.

We refer the reader also to [HHH23] which has independently and concurrently initiated the

¹Constant-cost randomized decision trees are equivalent to constant-cost deterministic decision trees, which compute functions $\{0, 1\}^n \rightarrow \{0, 1\}$ depending on only $O(1)$ variables [BD02], and constant-cost deterministic communication protocols compute $n \times n$ matrices which have only $O(1)$ distinct rows and columns.

study of constant-cost randomized communication, with an emphasis on the algebraic properties of these problems, including connections to operator theory and other areas. Since the preprint and conference version of the current paper [HWZ22], there have been many subsequent works extending and improving our results. We discuss these works in [Section 1.3](#).

1.1 Communication Complexity

Let us now discuss our results from the perspective of communication complexity. In [Section 1.2](#) we will discuss the results from the perspective of structural graph theory and graph representations.

We formally define the relevant notions of communication complexity in [Definition 2.1](#); here we write $R(P)$ for the minimum cost (on worst-case inputs) of a *public-coin* randomized communication protocol computing the problem P , and note that $R(P)$ is a function of the input domain size. We wish to understand the structure of problems which admit *constant-cost* randomized protocols, i.e., problems P with $R(P) = O(1)$. (Note that this goal would not be sensible for *private-coin* randomized protocols², since the constant-cost problems for this class of protocols are the same as the constant-cost problems for deterministic communication.) We organize the paper in order of studying Questions I-III.

Question I. What are natural examples of problems with constant cost?

EQUALITY is the standard example of a constant-cost problem, and more generally it has been known since at least [Yao03] that for any constant $k \in \mathbb{N}$, the k -HAMMING DISTANCE problem (where two players decide if their inputs $x, y \in \{0, 1\}^n$ differ on at most k bits) also has constant cost. But, to the best of our knowledge, almost no further non-trivial examples of constant-cost communication problems have appeared in the past 20 years. This makes it difficult to speculate on the general structure of constant-cost problems and the techniques required to design these “hyperefficient” protocols.

We present a number of new communication problems with non-trivial constant-cost communication protocols. Let us break these down into two sub-questions:

Question I (a). For which graphs G can two players decide if their vertices x, y are adjacent, or at distance at most k ? Every communication problem can be rephrased as: Alice and Bob each have a vertex in a shared graph G , and they want to decide if their vertices are adjacent³. One may also think of the k -HAMMING DISTANCE problem as: Alice and Bob have vertices of the hypercube graph, and wish to decide if their vertices have path-distance k . So, which other classes of graphs have constant-cost communication protocols for deciding adjacency or path-distance k ?

Our first example is that Alice and Bob can decide if their vertices have path-distance k in any *planar graph*, with communication cost depending on k but *not* on the number of vertices in the graph:

Theorem 1.1 (Informal; see [Section 4.1](#)). *Let k be any constant, and let Alice and Bob have vertices x, y in a shared planar graph G . Then there is a constant-cost randomized communication protocol for deciding $\text{dist}(x, y) \leq k$.*

This is a corollary of a more general result, using structural graph theory techniques from the literature on model checking. We generalize the problem in two ways:

²With bounded error. Constant-cost for private-coin *unbounded-error* protocols (as in [PS86]) is studied in a number of subsequent works; see [Section 1.3](#).

³The communication matrix may be interpreted as the adjacency matrix of a bipartite graph.

1. The statement $\text{dist}(x, y) \leq k$ can be written as a first-order formula $\phi(x, y)$ using the edge relation of the graph G , and with variables being vertices of the graph, i.e.:

$$\text{dist}(x, y) \leq k \equiv \phi(x, y) := (\exists v_1, \dots, v_k : x = v_1 \wedge y = v_k \wedge E(x_1, x_2) \wedge \dots \wedge E(x_{k-1}, x_k)).$$

We generalize the result to give communication protocols for deciding any such first-order formula.

2. We generalize the result to hold not only for planar graphs, but for any *stable* graph class of *bounded twin-width*. Including, for example, any proper minor-closed class or any class with bounded treewidth or bounded clique-width.

Question I (b). How can we generate new constant-cost problems? It is often helpful to be able to create new problems by composition. In graph theory terms, it is often helpful to understand what happens to graph properties under product operations. We show there is a general method to construct new constant-cost problems via a composition operation (which corresponds to the Cartesian product of graphs). This also leads to new techniques for constructing implicit graph representations for Cartesian product graphs.

Specifically, consider n independent instances P_1, \dots, P_n of any constant-cost problem \mathcal{P} , so that Alice and Bob have inputs x_1, \dots, x_n and y_1, \dots, y_n , respectively, to these problems. We may define a new problem $P_1 \square P_2 \square \dots \square P_n$ where Alice and Bob should output $P_i(x_i, y_i)$ if $i \in [n]$ is a unique index where they have received unequal inputs $x_i \neq y_i$ (otherwise they simply output 0). Note that, with only a budget of $O(1)$ communication, Alice and Bob cannot identify the index i where $x_i \neq y_i$; nevertheless, as we show, they can compute the output of this unknown instance. This generalizes the 1-HAMMING DISTANCE problem, which is obtained in this way when the “base problem” \mathcal{P} is simply the complement of the 2×2 identity matrix. Our proof that EQUALITY is not a complete problem (see Question III below) also shows that this composition takes any problem \mathcal{P} and turns it into a problem that does not reduce to EQUALITY. (Subsequent work [FGHH24] extends this composition method to find problems which do not reduce to any k -HAMMING DISTANCE.)

In graph theory terms, if \mathcal{F} is any graph class such that, for any graph $G \in \mathcal{F}$, two players can decide whether their vertices x, y are adjacent in G using a constant-cost protocol, then there exists also a constant-cost protocol for deciding adjacency of vertices in any Cartesian product graph $G^\square = G_1 \square G_2 \square \dots \square G_n$, where \square denotes the Cartesian product and each $G_i \in \mathcal{F}$. More generally, if deciding whether two vertices have path-distance k is a constant-cost problem in the class \mathcal{F} (as is the case for planar graphs), then deciding path-distance k in G^\square is also constant cost.

Theorem 1.2 (Section 4.2). *Suppose that for every constant k , there is a constant-cost randomized protocol for deciding $\text{dist}(x, y) \leq k$ in graphs $G \in \mathcal{F}$. Then for every constant k there is a constant-cost randomized protocol deciding $\text{dist}(x, y) \leq k$ in the class \mathcal{F}^\square of Cartesian products of graphs in \mathcal{F} .*

This allows a new method for constructing implicit representations of Cartesian product graphs, which were not known (see Section 1.2, prior work [CLR20], and subsequent work [EHZ24]).

Question II. What structures of a problem explain the existence or non-existence of a constant-cost protocol?

Constant-cost communication seems quite restrictive, and constant vs. non-constant is the simplest possible lower-bound question that one can ask. So, we may hope for a simple criterion to determine whether a communication problem has a constant-cost protocol or not. We observe in

Section 3.2 that constant-cost communication problems may always be expressed as a hereditary class \mathcal{F} of graphs (i.e. closed under taking induced subgraphs) where two players can decide adjacency of vertices x, y in a shared graph $G \in \mathcal{F}$, using a randomized communication protocol whose cost is independent of the number of vertices of G . Then the question becomes, which hereditary graph classes \mathcal{F} admit constant-cost randomized protocols for deciding adjacency in graphs $G \in \mathcal{F}$?

Necessary criterion 1: The number of n -vertex graphs in \mathcal{F} (a quantity sometimes called the *speed* of \mathcal{F}) must be at most $2^{O(n \log n)}$ (see Section 3.2). This is interesting because hereditary graph classes with $2^{\Theta(n \log n)}$ n -vertex graphs are already well-studied in structural graph theory and are sometimes called *factorial classes*. Here are two of the most natural ways to obtain hereditary factorial classes, which we will study in this paper:

1. Geometric intersection graphs. A class of graphs is obtained by associating each vertex with a geometric object (e.g. a line in \mathbb{R}^2) and making two vertices adjacent if and only if their associated objects intersect. These classes of graphs are hereditary by definition, and constant-dimensional geometric intersection graphs generally have factorial speed (see e.g. [Spi03]), so answering Question II requires understanding which subclasses of geometric intersection graphs admit constant-cost randomized protocols for deciding adjacency. Geometric intersection graphs are the subject of one of the main open problems about implicit graph representations [Spi03], and they are closely related to *unbounded error* randomized communication [PS86], making them vital for understanding randomized communication (see discussion in Section 1.3).
2. Classes defined via forbidden induced subgraphs. Every hereditary graph class may be obtained by choosing a set \mathcal{H} of graphs, and taking the class $\text{Free}(\mathcal{H})$ of graphs which do not contain any $H \in \mathcal{H}$ as an induced subgraph (see e.g. [KL15, Chapter 2]). For hereditary classes of *bipartite* graphs (such as the graph classes obtained from any constant-cost communication problem), it is known exactly which graph classes defined by forbidding exactly one graph H have factorial speed [All09, LZ17].

Necessary criterion 2: Stability. Although the above mentioned graph classes have factorial speed, they still may not have constant-cost communication protocols for deciding adjacency, because they are *unstable*, meaning that the GREATER-THAN problem can be reduced to the problem of deciding adjacency in these graphs. GREATER-THAN is a standard communication problem, where the two players receive numbers $i, j \in [N]$ and must decide whether $i > j$. This problem has randomized communication cost $\Theta(\log \log N)$ [Nis93, Vio15], which is exponentially superior than the deterministic cost $\Theta(\log N)$, but still non-constant. A graph class \mathcal{F} is *stable* if there is a fixed constant bound on the largest instance of GREATER-THAN which appears in the adjacency relation in \mathcal{F} .

Example 1.3. An interval graph is a geometric intersection graph where each vertex is represented by an interval $[a, b] \subseteq \mathbb{Z}$, and two vertices are adjacent when their intervals intersect. Two players holding vertices of an N -vertex interval graph cannot, in general, determine whether their vertices are adjacent using a constant-cost communication protocol, because one may easily construct arbitrarily large instances of GREATER-THAN inside the adjacency relation of interval graphs. But this does not yet characterize the non-existence of a constant-cost protocol for this problem, because *a priori* it is not clear whether there is another structure of the problem which would *also* prevent a constant-cost protocol from existing, even if we “remove” the GREATER-THAN subproblems.

This raises the question: For which factorial classes is *stability* both necessary and sufficient to

guarantee constant cost? For a surprising variety of examples (including interval graphs), we show that stability precisely characterizes the existence of constant-cost protocols. We prove:

Theorem 1.4 (Informal; [Section 5.1](#)). *For interval graphs and permutation graphs (another example of geometric intersection graphs), deciding adjacency between two vertices is a constant-cost problem if and only if the graphs are stable.*

Theorem 1.5 (Informal; [Section 5.2](#)). *For bipartite graph classes defined as above by forbidding a single bipartite graph H such that $\text{Free}(H)$ has factorial speed, deciding adjacency between vertices is again a constant-cost problem if and only if the graphs are stable.*

These results are also motivated by the connection to implicit graph representations, specifically [Question 1.9](#) discussed below. The proofs require new structural decompositions of interval graphs, permutation graphs, and H -free bipartite graphs, and provide further new examples of problems with constant-cost communication protocols. They also suggest (to us) a deeper relation between communication complexity and the stability condition. Stability plays an important role in structural graph theory and model theory (see [Section 5](#)). Its role in communication complexity remains unclear (see [Section 1.3](#)). Subsequent work has shown that stable factorial classes may not have constant-cost protocols [[HHH22](#), [EHK22](#)], and stability is crucial in recent lower bounds [[FHHH24](#)].

Question III. Is Equality *complete* for constant-cost randomized communication?

A useful way to try to characterize the constant-cost communication problems is to find a *complete* problem: a single constant-cost communication problem that all other constant-cost problems reduce to (see [Section 3.3](#) for the definition of reductions in the constant-cost setting). Depending on the properties of the complete problem, this could immediately answer many of the interesting questions about constant-cost randomized communication. EQUALITY is the first natural candidate that one might choose.

So, can every constant-cost communication protocol be rewritten in such a way that the only use of randomness in the protocol is the computation of EQUALITY instances, and the remainder of the protocol is completely deterministic? If this were so, we would know that the most extreme examples of the power of randomized communication are all explained by the power of the EQUALITY protocol, and we would know that to prove non-constant lower bounds for any problem, it suffices to prove lower bounds against *deterministic* protocols with access to the EQUALITY oracle.

However, we show that the answer is no: the 1-HAMMING DISTANCE problem is another problem that has a constant-cost protocol, but it cannot be reduced to EQUALITY.

Theorem 1.6 ([Section 6](#)). *EQUALITY is not complete for constant-cost randomized communication.*

A very different proof of this result appeared independently and concurrently in [[HHH23](#)], who use Fourier analysis to get optimal bounds on the number of EQUALITY oracle queries required to compute 1-HAMMING DISTANCE. Our proof is combinatorial and relies on a Ramsey-type theorem of [[ARSV06](#)], but it does not give quantitative bounds.

1.2 Implicit Graph Representations

We now discuss our results from the perspective of efficient graph representations. A central observation of this paper is that constant-cost randomized communication is the natural probabilistic version of *implicit graph representations* (see [Section 3.2](#)), which can be introduced with the following combinatorial question. Consider a hereditary graph class \mathcal{F} . How tightly can we “pack”

all the n -vertex graphs together? Or in other words, what is the smallest graph into which we can put all the n -vertex graphs in \mathcal{F} ? More formally, write \mathcal{F}_n for the set of graphs in \mathcal{F} with vertex set $[n]$. A *universal graph* (or *induced-universal graph*) for \mathcal{F} is a sequence $U = (U_n)_{n \in \mathbb{N}}$ of graphs, such that all n -vertex graphs $G \in \mathcal{F}_n$ appear as induced subgraphs of U_n . How big must U_n be?

This combinatorial question may also be expressed as an equivalent question in distributed computing and data structures, well-studied since [Mul89, KNR92]: For a graph $G \in \mathcal{F}$, we wish to represent G as efficiently as possible in an *implicit* or distributed manner: we would like to assign short binary labels $\ell(v)$ to each vertex v of G such that, given two labels $\ell(x), \ell(y)$, adjacency between x and y in G can be determined by a *decoder* which knows the class \mathcal{F} but not the specific graph G . This is an *adjacency labeling scheme*, and the existence of an adjacency labeling scheme with labels of size $s(n)$ for the n -vertex graphs is equivalent to the existence of a universal graph of size $2^{s(n)}$ [KNR92]. The main open problem is:

Question 1.7 (Implicit Graph Question (IGQ)). *Which hereditary graph classes admit a universal graph of size $\text{poly}(n)$ (i. e. an $O(\log n)$ -bit adjacency labeling scheme)?*

Adjacency labeling schemes with $O(\log n)$ bits are also known as *implicit representations*. This question has seen significant attention but little progress, although many positive examples are known. We introduce a probabilistic version of this question, which replaces adjacency labeling schemes with *randomized* adjacency labeling schemes, which we call *adjacency sketches*, and the equivalent notion of *probabilistic universal graphs (PUGs)*. Randomized adjacency labels for trees were introduced in [FK09], and have been studied more generally in connection to communication complexity by [Har20]; see [Definitions 3.1](#) and [3.2](#). We are interested in the classes that admit *constant-size* adjacency sketches (equivalently, constant-size PUGs), so we ask:

Question 1.8 (PUG Question). *Which hereditary graph classes admit a PUG of constant size (i. e., a constant-size adjacency sketch)?*

This question is interesting for three main reasons. First, we think it is both a natural question on its own, as well as being a natural probabilistic variant of the IGQ, and it is surprising that, as we will see, a rich variety of graph classes can be randomly projected into a *constant-size* universal graph.

Second, answering the PUG question is in fact *equivalent* to characterizing all communication problems with constant-cost randomized protocols. To see this, we make two observations (see [Section 3.2](#)):

- As mentioned above, any communication problem with a constant-cost randomized protocol is equivalent to a hereditary graph class with a constant-cost protocol for deciding adjacency.
- This constant-cost protocol for adjacency may be transformed into a constant-size adjacency *sketch*, or constant-size PUG. In the other direction, a constant-size adjacency sketch may be used as a constant-cost communication protocol for deciding adjacency.

Third, a simple derandomization argument shows that any class with a constant-size adjacency sketch also admits a size $O(\log n)$ adjacency labeling scheme, so the PUG question is an interesting special case of the IGQ, and it also leads to new techniques for constructing adjacency labeling schemes; for example, in [Question I \(b\)](#) above, we showed that Cartesian products preserve constant-size adjacency sketches, which imply new adjacency labeling schemes for Cartesian product graphs that were previously unknown (see e.g. [CLR20]). Since little progress has been made towards answering the IGQ, the PUG question may serve as an interesting step towards the IGQ, which makes stronger demands on the graph class. Since PUGs and adjacency sketches are stronger than

universal graphs and adjacency labeling schemes, one way to start understanding the relationship between the two is to ask when we can strengthen known adjacency labeling schemes to adjacency sketches; in other words:

Question 1.9. *Which hereditary graph classes with a universal graph of size $\text{poly}(n)$ also admit a constant-size PUG?*

This question motivates many of the examples we study in this paper (particularly interval and permutation graphs, and graphs of bounded twin-width). It is also related to Question III, since several standard examples of adjacency labeling schemes (e.g. for bounded-arboricity graphs [KNR92]) can in fact be rephrased as reductions to the EQUALITY problem. More generally, the most natural notion of a *reduction* between constant-cost communication problems (see Section 3.3) is equivalent to a natural notion of reduction between *graph classes*, for the purpose of studying adjacency labeling (which were also proposed independently in the literature on the IGQ in [Cha23]).

1.3 Subsequent Work

The concurrent work of Hambarzumyan, Hatami, & Hatami [HHH23] initiates the study of constant-cost communication from a different perspective and proposes a number of interesting conjectures about the structure of constant-cost problems. Following that paper and the conference version of the present work [HWZ22], there have been several follow-up works on communication complexity and implicit representations [HHH22, EHK22, EHZ24, HZ24], and numerous others related to constant-cost communication [HHP⁺22, ACHS23, CHHS23, HHM23, FHHH24, FGHH24] and adjacency sketching [NP24]; we briefly describe a few of these developments here, and we refer the reader to the subsequent works for discussions of open problems.

No complete problem. This paper and [HHH23] both show that the EQUALITY problem is not “complete” for the class of constant-cost randomized communication problems. Subsequent work [FHHH24] shows that in fact there is *no* complete problem.

New implicit representations, and generalized adjacency sketches. Extending the techniques in this paper, [EHZ24] prove optimal bounds on the size of universal graphs for any class of subgraphs of Cartesian product graphs, which improved upon the best-known bounds of earlier work [CLR20] and showed that they satisfy the IGQ. Recent work [NP24] has generalized the notion of adjacency sketching to *adversarial environments*.

Sign-rank and geometric intersection graphs. Informally, a family of matrices has *bounded sign-rank* if each matrix can be represented as a point-halfspace incidence matrix in a constant-dimensional space. Examples include the interval and permutation graphs studied in this article. A central open problem about implicit graph representations is whether all geometric intersection graphs (formally, the *semi-algebraic* graph classes) have universal graphs of size $\text{poly}(n)$. It is sufficient to consider graph classes whose adjacency matrices have bounded sign-rank (see e.g. [Spi03, Fit19, HZ24]). Understanding the matrices which have both bounded sign-rank and constant randomized communication cost has also emerged as one of the main open problems in constant-cost communication, one reason being that sign-rank is equivalent to *unbounded-error* randomized communication [PS86]; see [HHP⁺22, ACHS23, CHHS23, HHM23, HZ24]. So, bounded sign-rank is central to both implicit graph representations and constant-cost communication.

The implicit graph conjecture. As mentioned above, any hereditary graph class with a $\text{poly}(n)$ -size universal graph must contain at most $2^{O(n \log n)}$ distinct n -vertex graphs. A long-standing conjecture was that this condition is also *sufficient*:

Implicit graph conjecture [KNR92, Spi03]: A hereditary graph class has a universal graph of size $\text{poly}(n)$ (i.e., an $O(\log n)$ -size adjacency labeling scheme) if and only if it has at most factorial speed.

In the preprint of this paper, motivated by the idea that constant-size PUGs are the probabilistic version of $\text{poly}(n)$ -size universal graphs, we formulated a probabilistic version of the implicit graph conjecture. Based on the fact that stability plays a fundamental role in the structure of hereditary graph classes (e.g. it plays a role in certain “speed thresholds”, see [Appendix B](#), and is important in model theory and model checking), and the observation that, in each example considered in this paper, stability was the condition that determined whether constant-size PUGs exist, we proposed the following as a natural probabilistic variant of the implicit graph conjecture:

PUG Conjecture: A hereditary graph class has a constant-size PUG if and only if it is both stable and has at most factorial speed.

Shortly after the preprint became public, Hambardzumyan, Hatami, & Hatami refuted our PUG Conjecture [HHH22] using a construction from [HHH23], and Hatami & Hatami extended this refutation to refute the implicit graph conjecture itself [HH22]. This approach was further extended to refute variants of the conjecture for “small” graph classes (an important subfamily of factorial classes [BDS⁺24a]) and monotone graph classes [BDS⁺24b]. More examples refuting the PUG conjecture are now known, including (among others) the subgraphs of hypercubes [EHK22]. [EHK22] has resolved the PUG question ([Question 1.8](#)) for *monotone* graph classes.

1.4 Organization

The paper is organized in order of Questions I – III asked in [Section 1.1](#).

Section 2: Preliminaries. Notation and standard definitions of communication complexity, adjacency labeling, and universal graphs.

Section 3: Definitions and connections. Adjacency sketching, probabilistic universal graphs, the correspondence between communication complexity and hereditary graph classes, and stability.

Section 4, Question I – New examples. Two new examples of constant-cost communication (or constant-size adjacency sketches): (1) first-order formulae (including small distances) for planar graphs and their generalization, and (2) adjacency in Cartesian products of graphs.

Section 5, Question II – Structure and stability. Some examples where stability is both necessary and sufficient to guarantee constant-cost communication or constant-size PUGs: (1) subclasses of permutation graphs, (2) subclasses of interval graphs, and (3) subclasses of monogenic bipartite graph classes.

Section 6, Question III – Equality is not complete. A proof that the EQUALITY problem is not complete for constant-cost communication.

2 Preliminaries

We introduce in this section some notation and standard notions of communication complexity, adjacency labeling schemes, and universal graphs. Since this paper is intended for readers who may be unfamiliar with communication complexity or adjacency labeling schemes, we attempt to include all relevant definitions here.

2.1 Notation and Terminology

We write $\mathbb{1}[A]$ for the indicator of event A ; i.e., the function which is 1 if and only if statement A is true. For a finite set X , we write $x \sim X$ when x is a random variable drawn uniformly at random from X .

Graphs. All graphs in this work are simple, i.e., undirected, without loops and multiple edges. Let $G = (V, E)$ be a graph. We write $\text{ADJ}_G \in \{0, 1\}^{|V| \times |V|}$ for the adjacency matrix of G . For a subset $W \subseteq V$, we write $G[W]$ to denote the subgraph of G induced by W , i.e., a graph obtained from G by removing all vertices in $V \setminus W$. A graph H is an *induced subgraph* of G if $H = G[W]$ for some $W \subseteq V$. We write $H \sqsubset G$ to denote the fact that H is an induced subgraph of G . We also write \overline{G} for the *complement* of G , i.e., the graph (V, \overline{E}) where $(x, y) \in \overline{E}$ if and only if $(x, y) \notin E$. A set of pairwise non-adjacent vertices in a graph is an independent set, and a set of pairwise adjacent vertices is a clique. For two graphs $G = (V, E)$ and $H = (V', E')$ on disjoint vertex sets, the $G + H := (V \cup V', E \cup E')$ is called the *disjoint union* of G and H . We denote by K_n , P_n , and C_n , respectively a complete graph, a path, and a cycle, each on n vertices.

Bipartite graphs. A *colored* bipartite graph is a bipartite graph with a given bipartition of its vertex set. We denote a colored bipartite graph by a triple (X, Y, E) , where X, Y is the partition of its vertex set into two parts, and the function $E : X \times Y \rightarrow \{0, 1\}$ defines the edge relation. If a bipartite graph G is connected, it has a unique partition of its vertices into two parts and therefore there is only one colored bipartite graph corresponding to G ; (note that (X, Y, E) and (Y, X, E) are considered the same colored bipartite graph). If G is disconnected, however, there is more than one corresponding colored bipartite graph. The *bipartite adjacency matrix* of a colored bipartite graph $G = (X, Y, E)$ is the matrix $A \in \{0, 1\}^{|X| \times |Y|}$ with $A(i, j) = 1$ if and only if $(i, j) \in E$. Abusing notation, we also write ADJ_G for the bipartite adjacency matrix of G .

For colored bipartite graphs $G = (X, Y, E)$ and $H = (X', Y', E')$, we say that H is an induced subgraph of G , and write $H \sqsubset G$, when there is an injective map $\phi : X' \cup Y' \rightarrow X \cup Y$ that preserves adjacency and preserves parts. The latter means that the images $\phi(X')$ and $\phi(Y')$ satisfy either $\phi(X') \subseteq X, \phi(Y') \subseteq Y$ or $\phi(X') \subseteq Y, \phi(Y') \subseteq X$. A colored bipartite graph $G = (X, Y, E)$ is called *biclique* if every vertex in X is adjacent to every vertex in Y , and G is called *co-biclique* if $E = \emptyset$.

For a graph $G = (V, E)$ and two disjoint sets $X, Y \subseteq V$, we write $G[X, Y]$ for the colored bipartite graph (X, Y, E') where for $(x, y) \in X \times Y$, $(x, y) \in E'$ if and only if $(x, y) \in E$. A bipartite graph H is a *semi-induced subgraph* of G if there exist disjoint sets $X, Y \subseteq V$ such that $H = G[X, Y]$.

The *bipartite complement*, \overline{G} , of a colored bipartite graph $G = (X, Y, E)$ is the graph $\overline{G} := (X, Y, \overline{E})$ with $(x, y) \in \overline{E}$ if and only if $(x, y) \notin E$ for $x \in X, y \in Y$.

Classes of graphs and bipartite graphs. We define the following for graphs; the same discussion also applies to bipartite graphs. A *class* of graphs is a set of graphs that is closed under isomorphism. A class of graphs is *hereditary* if it is closed under taking induced subgraphs. It is

well-known that any hereditary graph class can be defined by its set of *minimal forbidden induced subgraphs* (see e.g. [KL15, Theorem 2.1.3]). That is, for any hereditary class \mathcal{F} , there is a *unique minimal* set of graphs \mathcal{H} such that \mathcal{F} is the class \mathcal{H} -free graphs, i.e., $\mathcal{F} = \text{Free}(\mathcal{H})$, where

$$\text{Free}(\mathcal{H}) := \{G : \forall H \in \mathcal{H}, H \not\subseteq G\}.$$

For a class of graphs \mathcal{G} , its *hereditary closure* $\text{cl}(\mathcal{G})$ is the set of all induced subgraphs of graphs in \mathcal{G} , i.e., $\text{cl}(\mathcal{G}) = \{H : \exists G \in \mathcal{G}, H \subseteq G\}$. Note, by definition, $\text{cl}(\mathcal{G})$ is a (minimal) hereditary graph class that includes \mathcal{G} .

2.2 Communication Complexity

We define some basic concepts in communication complexity and refer the reader to [NK96, RY20] for an introduction to communication complexity. Typically, a communication problem is a sequence $f = (f_n)_{n \in \mathbb{N}}$ of functions⁴ $f_n : [n] \times [n] \rightarrow \{0, 1\}$. In this paper it will be more convenient to define a *communication problem* as a set of (not necessarily square) matrices \mathcal{M} . For a fixed *communication matrix* $M \in \mathcal{M}$, we write $\text{R}(M)$ for the cost of the optimal two-way, randomized communication protocol computing M , defined as follows.

Informally, two players Alice and Bob share a source of randomness. Alice receives input x which is a row index of M , Bob receives input y which is a column index of M , and they communicate by sending messages back and forth using their shared randomness. After communication, Bob must output a (random) value $b \in \{0, 1\}$ such that $b = M(x, y)$ with probability at least $2/3$. The cost of such a protocol is the maximum, over all inputs x, y , of the number of bits communicated between the players. Formally, the definition is as follows.

Definition 2.1. A two-way public-coin communication protocol is a probability distribution \mathcal{D} over *communication trees*. For communication matrix $M \in \{0, 1\}^{m \times n}$, a communication tree T is a binary tree with each inner node being a tuple (p, μ) where $p \in \{A, B\}$ and either $\mu : [n] \rightarrow \{0, 1\}$ or $\mu : [m] \rightarrow \{0, 1\}$ depending on whether $p = A$ or $p = B$. Each edge of T is labeled either 0 or 1. Each leaf node is labeled either 0 or 1. For any fixed tree T and inputs $x \in [m], y \in [n]$, communication proceeds by setting the current node c to the root node. At each step of the protocol, if c is an inner node (A, μ) then Alice sends $\mu(x)$ to Bob and both players set c to the child along the edge labeled $\mu(x)$. If c is an inner node (B, μ) then Bob sends $\mu(y)$ to Alice, and both players set c to the child along the edge labeled $\mu(y)$. The protocol terminates when c becomes a leaf node, and the output is the value of the leaf node; we write $T(x, y)$ for the output of communication tree T on inputs x, y .

The distribution \mathcal{D} is required to satisfy the condition that, for all inputs x, y , $T(x, y) = M(x, y)$ with probability at least $2/3$ over the random choice of $T \sim \mathcal{D}$, and the *cost* of the protocol is the largest depth d of a tree T in the support of \mathcal{D} . We write $\text{R}(M)$ for the minimum cost of a protocol computing M .

For a communication problem \mathcal{M} , we denote by \mathcal{M}_n the subset of matrices $M \in \mathcal{M}$ with r rows and c columns such that $r + c = n$. Then we define the complexity $\text{R}(\mathcal{M})$ of the problem \mathcal{M} as the function

$$n \mapsto \max_{M \in \mathcal{M}_n} \text{R}(M).$$

2.3 Adjacency Labeling Schemes, Universal Graphs, and Factorial Classes

A *graph class* \mathcal{F} is a set of labeled graphs, closed under isomorphism. It is *hereditary* if it is also closed under vertex deletion (i.e. it is closed under taking induced subgraphs). For any graph class

⁴In the literature, the domain is usually $\{0, 1\}^n \times \{0, 1\}^n$. We use $[n] \times [n]$ to highlight the graph interpretation.

\mathcal{F} and any $n \in \mathbb{N}$, we write \mathcal{F}_n for the set of graphs $G \in \mathcal{F}$ with vertex set $[n]$.

Adjacency labeling schemes were introduced in [Mul89, KNR92] and are defined as follows.

Definition 2.2 (Adjacency labeling scheme). Let \mathcal{F} be a hereditary graph class. An *adjacency labeling scheme* for \mathcal{F} of size $s(n)$ consists of a *decoder* $D : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}$ such that, for all $n \in \mathbb{N}$ and all n -vertex graphs $G \in \mathcal{F}_n$, there exists a *labeling* $\ell_G : V(G) \rightarrow \{0, 1\}^{s(n)}$ of the vertices of G , where

$$\forall x, y \in V(G) : \quad \{x, y\} \in E(G) \iff D(\ell_G(x), \ell_G(y)) = 1.$$

A hereditary graph class which admits an adjacency labeling scheme of size $s(n) = O(\log n)$ is sometimes said to have an *implicit representation*, and characterizing the graph classes which have an implicit representation is the main open problem about adjacency labeling schemes, which we call the IGQ.

Universal graphs were introduced in [Rad64]. A universal graph (or induced-universal graph) is defined as

Definition 2.3. Let \mathcal{F} be a hereditary graph class. A *universal graph* for \mathcal{F} of size $s(n)$ is a sequence $U = (U_n)_{n \in \mathbb{N}}$ of graphs with $|U_n| = s(n)$, such that for all $n \in \mathbb{N}$ and all $G \in \mathcal{F}_n$, G is an induced subgraph of U_n .

As observed in [KNR92], these concepts are equivalent:

Proposition 2.4 ([KNR92]). *A hereditary graph class \mathcal{F} has an adjacency labeling scheme of size $\lceil s(n) \rceil$ if and only if it has a universal graph of size $2^{s(n)}$.*

Implicit representations, or adjacency labeling schemes of size $O(\log n)$, are therefore equivalent to universal graphs of size $\text{poly}(n)$. These graph classes satisfy a certain bound on the number of n -vertex graphs $|\mathcal{F}_n|$:

Proposition 2.5 ([KNR92]). *If a hereditary graph class \mathcal{F} has a universal graph of size $2^{O(\log n)} = \text{poly}(n)$ (i. e., an adjacency labeling scheme of size $O(\log n)$), then $|\mathcal{F}_n| \leq \binom{\text{poly}(n)}{n} = 2^{O(n \log n)}$.*

The function $n \mapsto |\mathcal{F}_n|$ is called the *speed* of the graph class. Graph classes with speed $2^{\Theta(n \log n)}$ are said to have *factorial speed*, and are called *factorial classes*. In [Appendix B](#) we summarize some known results about the family of factorial hereditary graph classes.

3 Connections: Randomized Communication and Implicit Representations

In this section we define the central concepts of the paper and prove some basic properties, including the connection between randomized communication and implicit representations.

Section 3.1: We define adjacency sketching and probabilistic universal graphs.

Section 3.2: We explain the correspondence between communication complexity, adjacency sketching, and implicit graph representations.

Section 3.3: We define notions of *reductions* between families of matrices, communication problems, and graph classes.

Section 3.4: We introduce some notation for *equality-based* communication and labeling, which will be used often throughout the paper.

Section 3.5: We state and prove some basic results about adjacency sketching, which will be used throughout the paper.

3.1 Adjacency Sketching and Probabilistic Universal Graphs

Probabilistic versions of adjacency labeling schemes and universal graphs were introduced in [Har20], which we call *adjacency sketches* and *probabilistic universal graphs (PUGs)*, respectively. A similar definition with one-sided error is given in [FK09]; see that paper for some results on the limitations of one-sided error. In the field of sublinear algorithms, a *sketch* reduces a large or complicated object to a smaller, simpler one that supports (approximate) queries. An adjacency sketch randomly reduces a hereditary graph class to a PUG that supports adjacency queries.

Definition 3.1 (Probabilistic Universal Graph). A *probabilistic universal graph sequence* for a graph class \mathcal{F} (or *probabilistic universal graph* for short) with error δ and size $m(n)$ is a sequence $U = (U_n)_{n \in \mathbb{N}}$ of graphs with $|U_n| = m(n)$ such that, for all $n \in \mathbb{N}$ and all $G \in \mathcal{F}_n$, the following holds: there exists a probability distribution over maps $\phi : V(G) \rightarrow V(U_n)$ such that

$$\forall u, v \in V(G) : \quad \mathbb{P}_{\phi} \left[\mathbb{1}[(\phi(u), \phi(v)) \in E(U_n)] = \mathbb{1}[(u, v) \in E(G)] \right] \geq 1 - \delta.$$

We say that a graph class \mathcal{F} admits a *constant-size PUG* if there exists a probabilistic universal graph sequence for \mathcal{F} with error $\delta = 1/3$ and size $m(n) = O(1)$.

Definition 3.2 (Adjacency Sketch). For a graph class \mathcal{F} , an *adjacency sketch* with size $c(n)$ and error δ is a pair of algorithms: a randomized *encoder* and a deterministic *decoder*. On input $G \in \mathcal{F}_n$, the encoder outputs a (random) function $\text{sk} : V(G) \rightarrow \{0, 1\}^{c(n)}$. The encoder and (deterministic) decoder $D : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}$ satisfy the condition that for all $G \in \mathcal{F}$,

$$\forall u, v \in V(G) : \quad \mathbb{P}_{\text{sk}} \left[D(\text{sk}(u), \text{sk}(v)) = \mathbb{1}[(u, v) \in E(G)] \right] \geq 1 - \delta.$$

In both definitions, we assume $\delta = 1/3$ unless otherwise specified. Setting $\delta = 0$ we obtain the (deterministic) labeling schemes of [KNR92]. We will write $\text{SK}(\mathcal{F})$ for the smallest function $c(n)$ such that there is an adjacency sketch for \mathcal{F} with size $c(n)$ and error $\delta = 1/3$.

PUGs are equivalent to adjacency sketches, by the same argument which shows that universal graphs are equivalent to adjacency labeling schemes [KNR92]: identify the vertices of U_n with the binary strings $\{0, 1\}^{c(n)}$ for $c(n) = \lceil \log |U_n| \rceil$, identify the random sketch $\text{sk} : V(G) \rightarrow \{0, 1\}^{c(n)}$ with the map $\phi : V(G) \rightarrow V(U_n)$, and identify the decoder D with the edge relation on U_n . We get:

Proposition 3.3. *A hereditary class \mathcal{F} has a constant-size PUG if and only if $\text{SK}(\mathcal{F}) = O(1)$.*

The choice of $\delta = 1/3$ in the definition of constant-size PUGs and adjacency sketches is arbitrary: by standard probability boosting arguments, one may achieve any constant error $\delta < 1/2$ while incurring only a constant-factor cost; we provide a proof for the sake of completeness in [Appendix A](#).

Proposition 3.4. *Let \mathcal{F} be a class of graphs. For any $\delta \in (0, 1/2)$, there is an adjacency sketch with error δ and size at most $O(\text{SK}(\mathcal{F}_n) \cdot \log \frac{1}{\delta})$. Equivalently, if there is a PUG $U = (U_n)_{n \in \mathbb{N}}$ for \mathcal{F} with size $|U_n|$, then there is a PUG $U' = (U'_n)_{n \in \mathbb{N}}$ for \mathcal{F} with error δ and $|U'_n| \leq |U_n|^{O(\log(1/\delta))}$.*

Adjacency sketches can be derandomized to obtain adjacency labeling schemes. This was observed in [Har20] as a consequence of a bound on the number of bits of randomness that are required, and can also be proved by simple derandomization; we give a proof for the sake of completeness in [Appendix A](#). It is sometimes required that adjacency labeling schemes be produced by an efficient algorithm, and we observe that efficiency is preserved by this derandomization:

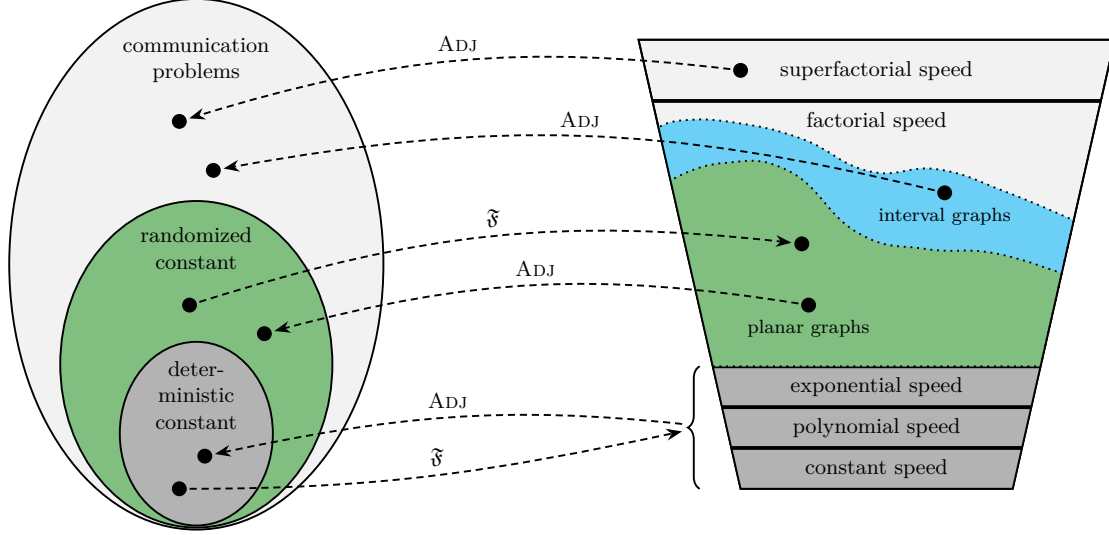


Figure 1: The correspondence that motivates this paper (Proposition 3.8). Appendix B describes the lattice on the right. Communication problems with constant-cost randomized protocols are mapped to the set of hereditary graph classes with constant-size PUGs (and therefore $\text{poly}(n)$ universal graphs by Proposition 3.6) by \mathfrak{F} . Families with constant-size PUGs are mapped to constant-cost communication problems by ADJ.

Proposition 3.5. *For any hereditary graph class \mathcal{F} , there is an adjacency labeling scheme of size $O(\text{SK}(\mathcal{F}) \cdot \log n)$. If there is a randomized algorithm which produces the adjacency sketch in time $\text{poly}(n)$, then there is a randomized algorithm which produces the adjacency labels in expected time $\text{poly}(n)$.*

It follows that hereditary graph classes which admit constant-size PUGs have the $\text{poly}(n)$ -vertex universal graphs, asked for by the IGQ, as illustrated in Figure 1.

Proposition 3.6. *If a hereditary class \mathcal{F} has a constant-size PUG (i. e., $\text{SK}(\mathcal{F}) = O(1)$) then it is a positive example to the IGQ (i. e., it admits a universal graph of size $\text{poly}(n)$).*

Remark 3.7. A further motivation for these objects from the perspective of communication complexity, given in [Har20], is that adjacency sketching (respectively, labeling) is a generalization of the randomized (resp. deterministic) *simultaneous message passing (SMP)* model of communication. In this generalization, the referee who must compute $f(x, y)$ from the messages of the players does not know f in advance; they only know a certain class \mathcal{F} which contains f , and the players must include sufficient information in their messages to compensate for the ignorance of the referee.

3.2 Communication-to-Graph Correspondence

We now explain that communication problems with constant-cost randomized communication protocols are equivalent to hereditary graph classes with constant-size PUGs. This correspondence is illustrated in Figure 1

Let \mathcal{M} be any communication problem, which we recall is a set of matrices. For any $M \in \mathcal{M}$ with $M \in \{0, 1\}^{n_1 \times n_2}$, we may consider the bipartite graph $G_M = ([n_1], [n_2], E)$ with edge $(i, j) \in E$ if and only if $M(i, j) = 1$. We define the hereditary class $\mathfrak{F}(\mathcal{M})$ associated with \mathcal{M} as the smallest hereditary class that contains each G_M . Formally,

$$\mathfrak{F}(\mathcal{M}) := \text{cl}(\{G_M : M \in \mathcal{M}\}).$$

One may equivalently think of $\mathfrak{F}(\mathcal{M})$ as being the communication problem containing every matrix which appears as a permutation of a submatrix of \mathcal{M} . It is important to keep in mind that, due to the fact that it is hereditary, $\mathfrak{F}(\mathcal{M})$ can have unintuitive consequences for the communication complexity: for example, some problems \mathcal{M} with small (but non-constant) randomized communication complexity can suddenly have maximum communication complexity when replaced with $\mathfrak{F}(\mathcal{M})$; see [Example 3.9](#).

In the other direction, for any set of graphs \mathcal{F} , we define the natural ADJACENCY communication problem, which captures the complexity of the two-player game of deciding whether the players are adjacent in a given graph $G \in \mathcal{F}$.

$$\text{ADJ}_{\mathcal{F}} := \{M \mid M \text{ is the adjacency matrix of some } G \in \mathcal{F}\}.$$

We may now state the formal equivalence between constant-cost communication and constant-size PUGs:

Proposition 3.8. *For any communication problem \mathcal{M} and hereditary graph class \mathcal{F} :*

1. \mathcal{F} has a constant-size PUG if and only if $R(\text{ADJ}_{\mathcal{F}}) = O(1)$.
2. $R(\mathcal{M}) = O(1)$ if and only if $\mathfrak{F}(\mathcal{M})$ has a constant-size PUG (i.e., $\text{SK}(\mathfrak{F}(\mathcal{M})) = O(1)$).

Proof. We start by proving [Item 1](#). First, suppose that \mathcal{F} is a hereditary graph class with $\text{SK}(\mathcal{F}) = O(1)$. Let D be the decoder of the constant-cost adjacency sketch, and for any graph $G \in \mathcal{F}$ write Φ_G for the distribution over sketches for G . We obtain a constant-cost communication protocol for $\text{ADJ}_{\mathcal{F}}$ as follows. Let $M \in \text{ADJ}_{\mathcal{F}}$ so that M is the adjacency matrix of some graph $G \in \mathcal{F}$ and we may think of the rows and columns of M as being indexed by vertices of G . On inputs $x, y \in V(G)$, Alice and Bob sample $\text{sk} \sim \Phi_G$ and Alice sends $\text{sk}(x)$ to Bob, which requires at most $\text{SK}(\mathcal{F})$ bits of communication. Then Bob simulates the decoder on $D(\text{sk}(x), \text{sk}(y))$ and sends the result to Alice. By definition

$$\mathbb{P}_{\text{sk} \sim \Phi_G} [D(\text{sk}(x), \text{sk}(y)) = M(x, y)] \geq 2/3.$$

Now suppose that $R(\text{ADJ}_{\mathcal{F}}) = O(1)$. Then there is a constant d such that for any $G \in \mathcal{F}_n$ it holds that the adjacency matrix $M \in \{0, 1\}^{n \times n}$ of G satisfies $R(M) \leq d$. For each $G \in \mathcal{F}$, let $\mathcal{P}(G)$ be the probability distribution over communication trees defined by an optimal communication protocol for the edge relation of G . Then it holds that every communication tree in the support of $\mathcal{P}(G)$ has depth at most d . We define the adjacency sketch for \mathcal{F} as follows. For every $G = (V, E) \in \mathcal{F}$, construct the random sketch sk by sampling $T \sim \mathcal{P}(G)$, and then for every $v \in V$:

For every node c of T , append to the label $\text{sk}(v)$ the following:

1. If c is an inner node (p, m) (with $p \in \{A, B\}$ and $m : [n] \rightarrow \{0, 1\}$), append the symbol p and the value $m(v)$.
2. If c is a leaf with value b , append the symbol L and the value b .

We define the decoder D as follows. On input $(\text{sk}(u), \text{sk}(v))$, the decoder simulates the communication tree T on (u, v) using the values $m(u), m(v)$ for each inner node. We therefore obtain

$$\mathbb{P}_{\text{sk}} [D(\text{sk}(u), \text{sk}(v)) = E(u, v)] = \mathbb{P}_{T \sim \mathcal{P}(G)} [T(u, v) = E(u, v)] \geq 2/3.$$

We now prove [Item 2](#). From the first argument above, it is clear that for any communication problem \mathcal{M} , if $\text{SK}(\mathfrak{F}(\mathcal{M})) = O(1)$ then $R(\mathcal{M}) = O(1)$. In the other direction, assume that $R(\mathcal{M}) \leq d$ for

some constant d , and consider $\text{SK}(\mathfrak{F}(\mathcal{M}))$. Then it holds for any $G \in \mathfrak{F}(\mathcal{M})$ that there exists $M \in \mathcal{M}$ such that the adjacency matrix A of G is a submatrix of M . Then $\text{R}(A) \leq \text{R}(M) \leq d$. We may then construct adjacency sketches by the scheme above, so we conclude $\text{SK}(\mathfrak{F}(\mathcal{M})) = O(1)$. \square

Note that this equivalence is special to constant-cost communication. The transformation of a communication problem to a hereditary graph class may have unintuitive consequences for problems with non-constant complexity, as in the next example.

Example 3.9. For a function $k(d)$, the $k(d)$ -HAMMING DISTANCE problem $\text{HD}^{k(d)}$ requires Alice and Bob to decide whether the Hamming distance between their inputs $x, y \in \{0, 1\}^d$ is at most $k(d)$. Note that this is a family of $n \times n$ matrices with $n = 2^d$. The randomized communication complexity of $\text{HD}^{k(d)}$ is $\Theta(k(d) \log k(d))$ when $k(d) = o(\sqrt{d})$ [HSZZ06, Saĝ18]. For example, if $k = \log \log d$ then the communication complexity is $\Theta((\log \log \log n) \cdot (\log \log \log n))$.

For any non-constant $k(d)$, no matter how small, the hereditary graph class $\mathcal{H} = \mathfrak{F}(\text{HD}^{k(d)})$ has a lower bound of $\text{R}(\text{ADJ}_{\mathcal{H}}) = \Theta(\log n)$ for n -vertex graphs, meaning that no protocol does better than simply sending the entire input to the other player. This is because, when we take the hereditary closure (essentially, we include every submatrix of a $k(d)$ -HAMMING DISTANCE problem for arbitrarily large d), we get the family of *all* bipartite graphs, as follows.

Suppose that $k(d)$ is increasing in d . Fix any n and let d be large enough that $d \geq 2n$ and $k(d) \geq n - 1$. Let $G = (X, Y, E)$ be any bipartite graph with $n = |X|$, and identify X with $[n]$. To each $x \in X$, assign the string $e_x \in \{0, 1\}^{2n}$ which is 0 everywhere except on coordinate $x \in [n]$. Now to each $y \in Y$, let $S \subseteq [n]$ be the set of its neighbors in G . Assign the string $a_y \in \{0, 1\}^{2n}$ which is 0 everywhere except on the coordinates in S , as well as on the last $n - |S|$ coordinates. Observe that the Hamming distance between e_x and a_y is $(n - |S|) + (|S| - 1) = n - 1$ if x is adjacent to y , and $(n - |S|) + (|S| + 1) = n + 1$ if x is not adjacent to y . We may now perform a simple padding operation to extend e_x to $e'_x \in \{0, 1\}^d$ and a_y to $a'_y \in \{0, 1\}^d$, such that the Hamming distance between e'_x, a'_y is $k(d)$ if x is adjacent to y and $k(d) + 2$ if x is not adjacent to y . So we have constructed a submatrix of the $k(d)$ -HAMMING DISTANCE matrix which is equal to the adjacency matrix of G .

3.2.1 Chain Number & Stability

We define the notion of *chain number*, which will be used in several of our proofs, and use it to formally define *stability*.

Definition 3.10 (Chain Number & Stability). For a graph G , the *chain number* $\text{ch}(G)$ is the maximum number k for which there exist disjoint sets of vertices $\{a_1, \dots, a_k\}, \{b_1, \dots, b_k\} \subseteq V(G)$ such that $(a_i, b_j) \in E(G)$ if and only if $i \leq j$. For a graph class \mathcal{F} , we write $\text{ch}(\mathcal{F}) = \max_{G \in \mathcal{F}} \text{ch}(G)$. If $\text{ch}(\mathcal{F}) = \infty$, then \mathcal{F} has *unbounded chain number*, otherwise it has *bounded chain number*. If \mathcal{F} has bounded chain number, we also call it *stable*.

The name *stable* for these classes is as in [CS18, NMP⁺21] (they are also called *graph-theoretically stable* in [GPT22]). These classes have many interesting properties, including stronger versions of Szemerédi’s Regularity Lemma [MS14] and the Erdős-Hajnal property [CS18] (it is also conjectured in [HHH23] that graphs which admit constant-cost protocols for adjacency satisfy the *strong* Erdős-Hajnal property), and they play a central role in algorithmic graph theory [GPT22]. We observe that stability is also essential for understanding the IGQ and randomized communication complexity. More specifically, for a hereditary graph class to have a constant-size PUG, it is necessary for it to be stable. This follows simply from known lower bounds on the GREATER-THAN communication problem (see [Appendix A.2](#)).

Proposition 3.11. *If a hereditary graph class \mathcal{F} is not stable, then $\text{SK}(\mathcal{F}) = \Omega(\log n)$.*

We conclude this section by noting a useful characterization of stable graph classes via forbidden induced subgraphs (Proposition B.4): a graph class \mathcal{F} has a bounded chain number (i.e., \mathcal{F} is stable) if and only if

$$\mathcal{F} \subseteq \text{Free}(H_p^{\circ\circ}, H_q^{\bullet\circ}, H_r^{\bullet\bullet}), \text{ for some choice of } p, q, r,$$

where $H_p^{\circ\circ}$ is a *half-graph*, $H_q^{\bullet\circ}$ is a *co-half-graph*, and $H_r^{\bullet\bullet}$ is a *threshold graph* (depicted in Figure 2).

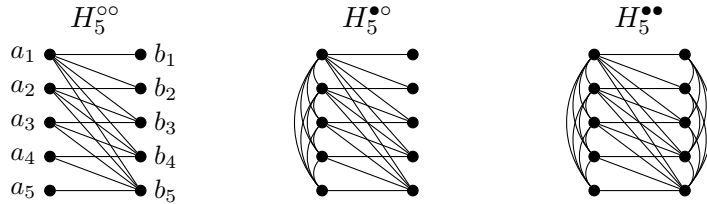


Figure 2: Examples of the half-graph, co-half-graph, and threshold graphs.

3.3 Reductions

We define a general notion of reductions useful for both communication complexity and implicit graph representations⁵. First, we define a notion of reduction between families of matrices. Given k matrices $B_1, B_2, \dots, B_k \in \{0, 1\}^{n \times n}$ and a Boolean function $h : \{0, 1\}^k \rightarrow \{0, 1\}$, we define

$$h(B_1, B_2, \dots, B_k) := A,$$

where $A \in \{0, 1\}^{n \times n}$ is the matrix with $A(i, j) = h(B_1(i, j), B_2(i, j), \dots, B_k(i, j))$ for all $i, j \in [n]$.

Definition 3.12 (Matrix Reductions). Let \mathcal{A}, \mathcal{B} be families of matrices. We say that \mathcal{A} *matrix-reduces to* \mathcal{B} if there exists a constant k such that, for every $A \in \mathcal{A}$, there exist $h : \{0, 1\}^k \rightarrow \{0, 1\}$ and $B_1, B_2, \dots, B_k \in \mathcal{B}$ so that

$$A = h(B_1, B_2, \dots, B_k).$$

3.3.1 Between communication problems

Informally, for communication problems \mathcal{A}, \mathcal{B} , we will say that \mathcal{A} has a *constant-cost reduction* to \mathcal{B} if \mathcal{A} can be computed by a constant-cost deterministic protocol with oracle access to \mathcal{B} . Formally, we define this as follows.

Definition 3.13 (Query Set). For any set of matrices \mathcal{Q} we define the *query set*

$$\text{QS}(\mathcal{Q}) := \{M \mid M \text{ is a blowup of a permutation of a submatrix of some } Q \in \mathcal{Q}\},$$

where we say that a matrix A is a *blowup* of B if it is obtained from B by a sequence of row and column duplications.

Definition 3.14 (Communication with Oracles). For any set \mathcal{Q} of matrices, we define the *deterministic \mathcal{Q} -oracle communication complexity* of the communication problem \mathcal{A} as follows. For any $M \in \mathcal{A}$ with row and columns indexed by $[n_1], [n_2]$ respectively, a \mathcal{Q} -oracle protocol for M is a

⁵This general notion of reductions did not appear in the conference version of this paper. It is an adaptation of the definitions presented in subsequent work [HZ24, FHHH24], which we include here for the sake of clarity, and consistency with subsequent work.

tree T whose leaves v are associated with output values $\ell(v) \in \{0, 1\}$, and whose inner nodes u are associated with $n_1 \times n_2$ matrices $Q_u \in \text{QS}(\mathcal{Q})$ (note that matrices of every size appear in $\text{QS}(\mathcal{Q})$ since it is closed under taking submatrices and blowups). On input pair $i \in [n_1], j \in [n_2]$, the protocol initializes a pointer p to the root of T , and in every round the protocol proceeds to

$$p \leftarrow \begin{cases} \text{left child of } p & \text{if } Q_p(i, j) = 0 \\ \text{right child of } p & \text{if } Q_p(i, j) = 1, \end{cases}$$

until p is a leaf, at which point it outputs $\ell(p)$, which is required to be equal to $M(i, j)$. We write $\text{D}^{\mathcal{Q}}(M)$ for the minimum cost (i.e., depth) of such a protocol, and we say $n \mapsto \text{D}^{\mathcal{Q}}(\mathcal{A}_n)$ is the deterministic \mathcal{Q} -oracle communication complexity of \mathcal{A} .

We may now define *constant-cost reductions* between communication problems.

Definition 3.15 (Constant-Cost Reductions). A communication problem \mathcal{A} *reduces* to (or has a *constant-cost reduction* to) \mathcal{B} if $\text{D}^{\mathcal{B}}(\mathcal{A}) = O(1)$.

Remark 3.16. This notion of reduction differs from the standard notion of an oracle reduction in communication complexity (e.g. [BFS86]) because it allows the oracle matrices to be submatrices of *arbitrarily large* instances of \mathcal{B} . Standard oracle reductions require the oracle matrix to be a submatrix of some $B \in \mathcal{B}$ with at most N rows and columns, where $\log N = \text{poly log}(\log n)$ to preserve communication complexities of the form $\text{poly log}(\log n)$ (where $\log n$ is the number of bits required to encode the inputs $i, j \in [n]$). Since we aim to preserve *constant-cost* communication, this requirement is unnecessary.

Proposition 3.17. *Suppose communication problem \mathcal{A} has a constant-cost reduction to problem \mathcal{B} , and $\text{R}(\mathcal{B}) = O(1)$. Then $\text{R}(\mathcal{A}) = O(1)$.*

Proof. The randomized communication cost of a matrix M is not changed by permutations or by row and column duplications, so $\text{R}(\text{QS}(\mathcal{B})) = O(1)$ also. Let $A \in \mathcal{A}$; we must show $\text{R}(A) = O(1)$. Since \mathcal{A} reduces to \mathcal{B} , there is a \mathcal{B} -oracle protocol for A , i.e., a communication tree T of constant depth $d = O(1)$, with each vertex v assigned a matrix $Q_v \in \text{QS}(\mathcal{B})$. We obtain a randomized communication protocol with cost $O(d \log d)$ by simulating the tree T on inputs x, y ; namely, for each node v of T we simulate the randomized protocol for Q_v with error $1/3d$, which can be done with communication cost $O(\log d)$ using standard error boosting. Since at most d oracle queries are simulated, the probability that the protocol has an error on any of them is at most $d \cdot 1/3d = 1/3$. \square

Constant-cost communication reductions are equivalent to matrix reductions as follows.

Proposition 3.18. *A communication problem \mathcal{A} has a constant-cost communication reduction to \mathcal{B} if and only if \mathcal{A} matrix-reduces to $\text{QS}(\mathcal{B})$.*

Proof. First, assume that \mathcal{A} matrix-reduces to $\text{QS}(\mathcal{B})$, i.e., there exists a constant k such that, for each $A \in \mathcal{A}$, there exist a function $h : \{0, 1\}^k \rightarrow \{0, 1\}$ and $Q_1, \dots, Q_k \in \text{QS}(\mathcal{B})$ such that

$$A = h(Q_1, Q_2, \dots, Q_k).$$

Then for any $A \in \mathcal{A}$, we obtain $\text{D}^{\mathcal{B}}(A) \leq k$ by using the following communication protocol: on inputs x, y , the players query $Q_1(x, y), Q_2(x, y), \dots, Q_k(x, y)$ and then output $h(Q_1(x, y), \dots, Q_k(x, y))$.

Now, assume that $\text{D}^{\mathcal{B}}(\mathcal{A}) = O(1)$. Then for any $A \in \mathcal{A}$, there is an oracle communication tree T with constant depth $d = O(1)$ and each inner node v assigned a matrix $Q_v \in \text{QS}(\mathcal{B})$. Let v_1, \dots, v_t be inner nodes of T , where $t \leq 2^d$. Then we may write

$$A = h(Q_{v_1}, Q_{v_2}, \dots, Q_{v_t}),$$

where $h : \{0, 1\}^t \rightarrow \{0, 1\}$ is the function which simulates T given the answers to every query. \square

3.3.2 Between graph classes

Let \mathcal{F}, \mathcal{G} be hereditary classes of graphs. We define a reduction between \mathcal{F}, \mathcal{G} with the purpose of preserving the existence of implicit representations.

Let $\text{ADJ}_{\mathcal{F}}$ denote the set of adjacency matrices of graphs in \mathcal{F} . We say \mathcal{F} reduces to \mathcal{G} if $\text{ADJ}_{\mathcal{F}}$ matrix-reduces to $\text{QS}(\text{ADJ}_{\mathcal{G}})$, where $\text{QS}(\cdot)$ denotes the query set defined above. In graph-theoretic terms, $\text{QS}(\text{ADJ}_{\mathcal{G}})$ is the set of bipartite adjacency matrices obtained as follows: for any graph $G \in \mathcal{G}$, let $X, Y \subset V(G)$ be any subsets of vertices and $H = (X, Y, E_H)$ be the bipartite graph where $(x, y) \in X \times Y$ is an edge of H if and only if $\{x, y\}$ is an edge of G . Then take any $H' = (X', Y', E'_H)$ obtained from H by duplicating vertices. $\text{QS}(\text{ADJ}_{\mathcal{G}})$ is the set of bipartite adjacency matrices of any graph obtained in this way.

Proposition 3.19. *Suppose that \mathcal{F} reduces to \mathcal{G} and \mathcal{G} admits an adjacency labeling scheme of size $s(n)$. Then \mathcal{F} admits an adjacency labeling scheme of size $O(s(n))$.*

Proof. First, observe that, if \mathcal{G} admits an adjacency labeling scheme of size $s(n)$, then the family \mathcal{H} of bipartite graphs whose adjacency matrices are in $\text{QS}(\text{ADJ}_{\mathcal{G}})$ admits an adjacency labeling scheme of size $s(n) + 1$: to every vertex $v \in X \cup Y$ of a graph $H = (X, Y, E_H)$ constructed from $G \in \mathcal{G}$ as in the paragraph above (without vertex duplication), simply take the original label for the scheme of G and append one bit to indicate whether $v \in X$ or $v \in Y$. For the graph $H' = (X', Y', E'_H)$ obtained by duplicating vertices of H , simply copy the corresponding labels. It is straightforward to verify that such labels contain sufficient information to correctly deduce the adjacency relation between vertices in graphs from \mathcal{H} .

Now, there exists a constant k such that, for any graph $G \in \mathcal{F}$, there exist bipartite graphs $B_1, \dots, B_k \in \mathcal{H}$ with bipartite adjacency matrices $\text{ADJ}_{B_i} \in \text{QS}(\text{ADJ}_{\mathcal{G}})$ and a function $h : \{0, 1\}^k \rightarrow \{0, 1\}$, such that

$$\text{ADJ}_G = h(\text{ADJ}_{B_1}, \text{ADJ}_{B_2}, \dots, \text{ADJ}_{B_k}).$$

If \mathcal{F} is a class of bipartite graphs, then for $G = (X, Y, E)$, ADJ_G has rows indexed by X and columns indexed by Y , and we may assume that each ADJ_{B_i} also has rows indexed by X and columns indexed by Y . We label each $v \in X \cup Y$, with the concatenation of the encoding of h and the labels of v for each B_i in the labeling scheme for \mathcal{H} . Then the decoder can decide if $u, w \in X \cup Y$ are in different parts of the graph, and if so, decide adjacency of u and w in each B_i and compute h on these adjacency values. The size of the labels is at most $2^k + k(s(n) + 1) = O(s(n))$ since k is constant.

Now, if \mathcal{F} is not a class of bipartite graphs, then for $G = (V, E)$, ADJ_G is symmetric and has rows and columns indexed by V . Each B_i is a bipartite graph with the parts V and V' , a copy of V . For each $v \in V$, we assign a label by concatenating the encoding of h with two labels $a(v), b(v)$, where $a(v)$ is the concatenation of the labels of $v \in V$ in each B_i , and $b(v)$ is the concatenation of the labels of $v \in V'$ in each B_i . Given two labels for $u, v \in V$, the decoder can determine adjacency in G by means of the function h and the decoder for the labeling scheme of \mathcal{H} on each B_i , using the relevant parts of $a(v)$ and $b(u)$. The size of the labels is at most $2^k + 2k(s(n) + 1) = O(s(n))$. \square

Remark 3.20. It may seem more natural to define reductions by saying \mathcal{F} reduces to \mathcal{G} if $\text{ADJ}_{\mathcal{F}}$ matrix-reduces to $\text{ADJ}_{\mathcal{G}}$ instead of $\text{QS}(\text{ADJ}_{\mathcal{G}})$. This would preserve more of the graph structure through the reduction, since $\text{QS}(\text{ADJ}_{\mathcal{G}})$ allows taking semi-induced bipartite subgraphs (instead of induced subgraphs) and allows row and column duplication (i.e., duplicating vertices). But here we are interested only in preserving existence of adjacency labeling, and the reduction defined here is more permissive.

3.4 Equality-Based Labeling Schemes

We define a certain type of adjacency labeling scheme called an *equality-based labeling scheme*. A number of our results will be proved by constructing equality-based labeling schemes, so we will introduce some special notation for these. For simplicity of notation, we write

$$\text{EQ}(a, b) = \mathbb{1}[a = b].$$

Definition 3.21. (Equality-based Labeling Scheme). Let \mathcal{F} be a class of graphs. An (s, k) -equality-based labeling scheme for \mathcal{F} is a labeling scheme defined as follows. For every $G \in \mathcal{F}$ with vertex set $[n]$ and every $x \in [n]$, the label $\ell(x)$ consists of the following:

1. A prefix $p(x) \in \{0, 1\}^s$. If $s = 0$ we write $p(x) = \perp$.
2. A sequence of k equality codes $q_1(x), \dots, q_k(x) \in \mathbb{N}$.

The decoder must be of the following form. There is a set of functions $D_{p_1, p_2} : \{0, 1\}^{k \times k} \rightarrow \{0, 1\}$ defined for each $p_1, p_2 \in \{0, 1\}^s$ such that, for every $x, y \in [n]$, it holds that $(x, y) \in E(G)$ if and only if $D_{p(x), p(y)}(Q_{x,y}) = 1$, where $Q_{x,y} \in \{0, 1\}^{k \times k}$ is the matrix with entries $Q_{x,y}(i, j) = \mathbb{1}[q_i(x) = q_j(y)]$. If $s = 0$ we simply write $D(Q_{x,y})$.

We say that \mathcal{F} admits a *constant-size equality-based* labeling scheme if there exist constants s, k such that \mathcal{F} admits an (s, k) -equality-based labeling scheme.

Remark 3.22. We will often use the following notation. A label for x will be written as a constant-size tree of tuples of the form

$$(p_1(x), \dots, p_r(x) \mid q_1(x), \dots, q_t(x)),$$

where the symbols $p_i(x)$ belong to the prefix, while the symbols $q_i(x)$ are equality codes. When r, t are constants, it is straightforward to put such a label into the form required by [Definition 3.21](#).

Proposition 3.23. *Suppose a class of graphs \mathcal{F} admits an (s, k) -equality based labeling scheme. Then $\text{SK}(\mathcal{F}) = O(s + k \log k)$.*

Proof. Let $G \in \mathcal{F}$ and let n be the number of vertices. Let $T \subseteq \mathbb{N}$ be the set of equality codes $T = \{q_i(x) : i \in [k], x \in V(G)\}$. We assign the sketches to $V(G)$ as follows. For each $t \in T$ we assign a uniformly random $b(t) \in [3k^2]$ and observe that $b(t)$ may be encoded in $O(\log k)$ bits. To each vertex x we assign the sketch containing the s bits of the prefix $p(x)$, followed by $b(q_1(x)), \dots, b(q_k(x))$, which requires $O(s + k \log k)$ bits.

Consider any two vertices x, y . The decoder, upon receiving the sketches for x and y , constructs the matrix $Q'_{x,y}$ defined as $Q'_{x,y}(i, j) = \text{EQ}(b(q_i(x)), b(q_j(y)))$, and then applies the decoder $D_{p(x), p(y)}(Q'_{x,y})$ from the equality-based labeling scheme. Observe that $\text{EQ}(b(q_i(x)), b(q_j(y))) = \text{EQ}(q_i(x), q_j(y))$ with probability at least $1 - \frac{1}{3k^2}$. Then by the union bound, we have $Q'_{x,y} = Q_{x,y}$ with probability at least $2/3$, in which case the decoder outputs the correct value. \square

Example 3.24. Many standard examples of adjacency labeling schemes are in fact equality-based labeling schemes. For example, we may write the original adjacency labeling scheme of [\[KNR92\]](#) for forests as an equality-based labeling scheme: the label for any vertex $x \in [n]$ is

$$(- \mid x, p(x)),$$

where the first equality code is the vertex itself, and the second equality code is its parent $p(x)$. Given labels $(- \mid x, p(x))$ and $(- \mid y, p(y))$, the decoder outputs $(x = p(y)) \vee (y = p(x))$.

Let us now define equivalence graphs.

Definition 3.25. A graph G is an *equivalence graph* if it is a disjoint union of complete graphs. A colored bipartite graph $G = (X, Y, E)$ is a *bipartite equivalence graph* if it is a colored disjoint union of bicliques, i.e., if there are partitions $X = X_1 \cup \dots \cup X_m$, $Y = Y_1 \cup \dots \cup Y_m$ such that each $G[X_i, Y_i]$ is a biclique and each $G[X_i, Y_j]$ is a co-biclique when $i \neq j$.

Alternatively, the equivalence graphs are exactly the P_3 -free graphs, and the bipartite equivalence graphs are exactly the P_4 -free bipartite graphs.

Equality-based labeling schemes are related to reductions by the following proposition. For the sake of completeness, we state the simple proof in [Appendix A.3](#).

Proposition 3.26. *The following are equivalent for a hereditary graph class \mathcal{F} :*

1. \mathcal{F} admits a constant-size equality-based labeling scheme;
2. The graph class \mathcal{F} reduces to the class of equivalence graphs;
3. The communication problem $\text{ADJ}_{\mathcal{F}}$ reduces to EQUALITY.

Therefore, if \mathcal{F} admits a constant-size equality-based labeling scheme, it admits a constant-size adjacency sketch (and hence a constant-size PUG).

3.5 Basic Adjacency Sketches

Adjacency sketches for graph classes of bounded arboricity will be basic building blocks for some of our results.

Definition 3.27. A graph $G = (V, E)$ has *arboricity* α if its edges can be partitioned into at most α forests.

The classic adjacency labeling scheme of [KNR92] for graph classes of bounded arboricity may be interpreted as an $(0, \alpha)$ -equality-based labeling scheme (as in [Example 3.24](#)). Using [Proposition 3.23](#), this gives an adjacency sketch of size $O(\alpha \log \alpha)$, which was also stated in [Har20]. We show that this can be improved as follows:

Lemma 3.28. *For any $\alpha \in \mathbb{N}$, let \mathcal{A} be the class of graphs with arboricity at most α . Then \mathcal{A} admits a constant-size equality-based adjacency labeling scheme. \mathcal{A} also admits an adjacency sketch of size $O(\alpha)$.*

Proof. For any graph $G \in \mathcal{A}_n$ with vertex set $[n]$, partition the edges of G into forests F_1, \dots, F_α and to each tree in each forest, identify some arbitrary vertex as the root. For every vertex x , assign equality codes $q_1(x) = x$ and for $i \in [\alpha]$ set $q_{i+1}(x)$ to be the parent of x in forest F_i ; if x is the root assign $q_{i+1}(x) = 0$. For vertices x, y , the decoder outputs

$$\left(\bigvee_{j=2}^{\alpha} \mathbb{1}[q_1(x), q_j(y)] \right) \vee \left(\bigvee_{j=2}^{\alpha} \mathbb{1}[q_1(y), q_j(x)] \right).$$

This is 1 if and only if y is the parent of x or x is the parent of y in some forest F_i .

One can apply [Proposition 3.23](#) to obtain an $O(\alpha \log \alpha)$ adjacency sketch. We can improve this using a Bloom filter, since the output is simply a disjunction of equality checks. To each $i \in [n]$, assign a uniformly random number $r(i) \sim [6\alpha]$, and to each vertex x assign the sketch $(r(x), b(x))$ where $b(x) \in \{0, 1\}^{6\alpha}$ satisfies $b(x)_i = 1$ if and only if $r(q_j(x)) = i$ for some $j \in \{2, \dots, \alpha + 1\}$. On

input $(r(x), b(x))$ and $(r(y), b(y))$, the decoder outputs 1 if and only if $b(x)_{r(y)} = 1$ or $b(y)_{r(x)} = 1$. If y is a parent of x in any of the α forests, then $y = q_j(x)$ for some j , so $b(x)_{r(y)} = b(x)_{r(q_j(x))} = 1$ and the decoder will output 1 with probability 1. Similarly, if x is a parent of y in any of the α forests, the decoder will output 1 with probability 1. The decoder fails only when x, y are not adjacent and $r(x) = r(q_j(y))$ or $r(y) = r(q_j(x))$ for some j . By the union bound, this occurs with probability at most $2\alpha \cdot \frac{1}{6\alpha} = 1/3$, as desired. The size of the sketches is $O(\alpha + \log \alpha) = O(\alpha)$. \square

Remark 3.29. Graph classes of bounded arboricity include many commonly-studied hereditary graph classes, e.g., classes of graphs of bounded degree, classes of bounded treewidth, and proper minor-closed classes [Mad67].

4 Question I. New Examples of Constant-Cost Communication

This first main section of results is dedicated to finding new examples of problems with constant-cost randomized communication, to better understand the variety of problems in this class; this also leads to some new methods for constructing adjacency labeling schemes.

Section 4.1. We study the problem of computing path distance $\text{dist}(x, y) \leq k$ in graphs, where the players are given vertices x, y of a shared graph. It has long been known [Yao03, HSZZ06] that the k -HAMMING DISTANCE problem has a constant-cost randomized communication protocol whenever k is constant, which we may think of as computing path distance $\text{dist}(x, y) \leq k$ for vertices x, y in the hypercube graph, with cost independent of the number of vertices. One of the questions motivating this paper is, for which other graphs can we compute path distance $\text{dist}(x, y) \leq k$ in randomized communication cost independent of the number of vertices? Equivalently, for which hereditary classes of graphs can we construct constant-size *sketches* for deciding small distances? We generalize this problem to constructing constant-size sketches for deciding certain first-order formulae $\phi(x, y)$ over vertices x, y .

Section 4.2. We show that constant-size adjacency sketches are preserved by the Cartesian product operation on graphs. This also gives a new method of constructing adjacency labeling schemes for Cartesian products.

4.1 Computing Small Distances and First-Order Formulae

Graph width parameters like treewidth [RS86] and clique-width [CER93] are a central tool in structural graph theory. Often, graph classes where a certain width parameter is bounded possess favorable structural, algorithmic, or combinatorial properties. The *twin-width* parameter, introduced recently in [BKTW20], generalizes treewidth and clique-width and has attracted a lot of recent attention [GPT22, SS22, AHKO22, BH21]. Graph classes of bounded twin-width admit size $O(\log n)$ adjacency labeling schemes [BGK⁺21], making them a natural choice for studying [Question 1.8](#). Building upon recent structural results on stable classes of bounded twin-width [GPT22], we prove:

Theorem 4.1. *Let \mathcal{F} be a hereditary graph class of bounded twin-width. Then \mathcal{F} admits a constant-size PUG if and only if \mathcal{F} is stable.*

We also consider a type of sketches that generalizes adjacency and distance- k sketches. For any *first-order formula* $\phi(x, y)$ on vertex pairs (see [Section 4.1](#)), we consider sketches that allow to

decide $\phi(x, y)$ instead of adjacency. An example is the $\text{dist}(x, y) \leq k$ formula:

$$\delta_k(x, y) := (\exists v_1, v_2, \dots, v_{k-1} : (E(x, v_1) \vee x = v_1) \wedge (E(v_1, v_2) \vee v_1 = v_2) \wedge \dots \wedge (E(v_{k-1}, y) \vee v_{k-1} = y)).$$

Using results on *first-order transductions* (a graph transformation that arises in model theory) and their relation to stability and twin-width [BKTW20, NMP⁺21], we obtain the following corollary of Theorem 4.1.

Corollary 4.2. *Let \mathcal{F} be a stable class of bounded twin-width and let $\phi(x, y)$ be a first-order formula. Then \mathcal{F} admits a constant-size sketch for deciding ϕ .*

This implies Theorem 1.1 and its generalization stated in the introduction; it gives us constant-size distance- k sketches for stable classes of bounded twin-width, and, in particular, answers a question of [Har20], who asked about distance- k sketches for planar graphs (which are stable, because they have bounded arboricity, and are of bounded twin-width [BKTW20]). Our results have subsequently been extended and improved in [EHK22].

In Section 4.1.1 we provide necessary definitions and notations. To prove the theorem, we will first reduce the problem to bipartite graphs in Section 4.1.2, and then show in Section 4.1.3 how to construct a constant-size equality-based labeling scheme for any stable class of bipartite graphs of bounded twin-width. In Section 4.1.4 we will generalize the above theorem to first-order labeling schemes to show Corollary 4.2.

4.1.1 Preliminaries

Let $G = (V, E)$ be a graph. A pair of disjoint vertex sets $X, Y \subseteq V$ is *pure* if either $\forall x \in X, y \in Y$ it holds that $(x, y) \in E$, or $\forall x \in X, y \in Y$ it holds that $(x, y) \notin E$.

Definition 4.3 (Twin-Width). An *uncontraction sequence of width d* of a graph $G = (V, E)$ is a sequence $\mathcal{P}_1, \dots, \mathcal{P}_m$ of partitions of V such that:

- $\mathcal{P}_1 = \{V\}$;
- \mathcal{P}_m is a partition into singletons;
- For $i = 1, \dots, m - 1$, \mathcal{P}_{i+1} is obtained from \mathcal{P}_i by splitting exactly one of the parts into two;
- For every part $U \in \mathcal{P}_i$ there are at most d parts $W \in \mathcal{P}_i$ with $W \neq U$ such that (U, W) is not pure.

The *twin-width* $\text{tw}(G)$ of G is the minimum d such that there is an uncontraction sequence of width d of G .

The following fact, that we need for some of our proofs, uses the notion of first-order (FO) transduction. We omit the formal definition of FO transductions and refer the interested reader to e.g. [GPT22]. Informally, given a first-order formula $\phi(x, y)$ and a graph G , a *first-order (FO) ϕ -transduction* of G is a transformation of G to a graph that is obtained from G by first taking a constant number of vertex-disjoint copies of G , then coloring the vertices of the new graph by a constant number of colors, then using ϕ as the new adjacency relation, and finally taking an induced subgraph. An FO ϕ -transduction of a graph class \mathcal{F} is a class of ϕ -transductions of graphs in \mathcal{F} . A graph class \mathcal{G} is an FO-transduction of \mathcal{F} if there exists an FO formula ϕ so that \mathcal{G} is an ϕ -transduction of \mathcal{F} .

Theorem 4.4 ([BKTW20, NMP⁺21]). *Let \mathcal{F} be a stable class of bounded twin-width. Then any FO transduction of \mathcal{F} is also a stable class of bounded twin-width.*

4.1.2 From General Graphs to Bipartite Graphs

We begin by defining a natural mapping $\text{bip}(G)$ that transforms a graph G into a bipartite graph.

Definition 4.5. For any graph $G = (V, E)$ we define the colored bipartite graph $\text{bip}(G) = (V, V^\triangleleft, E^\triangleleft)$, where V^\triangleleft is a copy of V , as follows. For each $v \in V$ let $\triangleleft v \in V^\triangleleft$ denote its copy in V^\triangleleft . Then for each $x, y \in V$ we have $E(x, y) = E(y, x) = E^\triangleleft(x, \triangleleft y) = E^\triangleleft(y, \triangleleft x)$. For any set \mathcal{F} of graphs, we define

$$\text{bip}(\mathcal{F}) := \{\text{bip}(G) : G \in \mathcal{F}\}.$$

We require the following two simple properties of this transformation.

Proposition 4.6. *Let \mathcal{F} be a class of graphs. Then the class $\text{cl}(\text{bip}(\mathcal{F}))$ is an FO transduction of \mathcal{F} .*

Proof sketch. Let $G = (V, E)$ be a graph in \mathcal{F} . To obtain the graph $\text{bip}(G)$, we take a disjoint union $G_1 \cup G_2$, where $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are two copies of G , and transform this graph using an FO formula $\phi(x, y)$ that does not hold for any pair of vertices that belong to the same copy of G , and holds for any pair of vertices that are in different copies and whose preimages are adjacent in G :

$$\phi(x, y) = [(x \in V_1 \wedge y \in V_2) \vee (y \in V_1 \wedge x \in V_2)] \wedge \exists y' : M(y, y') \wedge (E_1(x, y') \vee E_2(x, y')),$$

where M is a relation that is true exactly for copies of the same vertex. For any induced subgraph of $\text{bip}(G)$, we in addition take an induced subgraph. \square

Proposition 4.7. *Let \mathcal{F} be a hereditary graph class.*

- (1) *If $\text{cl}(\text{bip}(\mathcal{F}))$ has an adjacency labeling scheme of size $s(n)$, then \mathcal{F} has an adjacency labeling scheme of size $O(s(2n))$.*
- (2) *If $\text{cl}(\text{bip}(\mathcal{F}))$ has a constant-size equality-based labeling schemes, then so does \mathcal{F} .*

Proof sketch. For $G \in \mathcal{F}$, we may assign labels to $v \in V(G)$ by concatenating the labels for v, v^\triangleleft in $\text{bip}(G)$, which are of size at most $s(2n)$. Given the labels for $(u, u^\triangleleft), (v, v^\triangleleft)$, the decoder can determine adjacency from the labels of u, v^\triangleleft . Thus, the total label size is at most $2s(2n) = O(s(2n))$. A similar argument works for the second part of the statement. \square

We now see that to prove [Theorem 4.1](#), it is enough to establish its special case for the classes of bipartite graphs. Assume [Theorem 4.1](#) holds for the bipartite graph classes and let \mathcal{F} be an arbitrary hereditary stable class of graphs of bounded twin-width. Then, by [Proposition 4.6](#) and [Theorem 4.4](#), the class $\text{cl}(\text{bip}(\mathcal{F}))$ is also stable and has bounded twin-width. Therefore, by [Proposition 4.7](#), any constant-size equality-based labeling scheme for $\text{cl}(\text{bip}(\mathcal{F}))$ can be turned into a constant-size equality-based labeling scheme for \mathcal{F} .

4.1.3 Bipartite Graphs of Bounded Twin-width

An *ordered graph* is a graph equipped with a total order on its vertices. We will denote ordered graphs as (V, E, \leq) and bipartite ordered graphs as (X, Y, E, \leq) , where \leq is a total order on V and on $X \cup Y$ respectively. A *star forest* is a graph whose every component is a star. Let (X, \leq) be a totally-ordered set. A subset $S \subseteq X$ is *convex* if for every $x, y, z \in X$ with $x \leq y \leq z$, such that $x, z \in S$, it holds also that $y \in S$.

Definition 4.8 (Division). A *division* of an ordered bipartite graph $G = (X, Y, E, \leq)$ is a partition \mathcal{D} of $X \cup Y$ such that each part $P \in \mathcal{D}$ is convex and either $P \subseteq X$ or $P \subseteq Y$. We will write $\mathcal{D}^X = \{P \in \mathcal{D} : P \subseteq X\}$, $\mathcal{D}^Y = \{P \in \mathcal{D} : P \subseteq Y\}$.

Definition 4.9 (Quotient Graph). For any ordered bipartite graph $G = (X, Y, E, \leq)$ and any division \mathcal{D} , the *quotient graph* G/\mathcal{D} is the bipartite graph $(\mathcal{D}^X, \mathcal{D}^Y, \mathcal{E})$ where $A \in \mathcal{D}^X, B \in \mathcal{D}^Y$ are adjacent if and only if there exist $x \in A, y \in B$ such that (x, y) are adjacent in G .

Definition 4.10 (Convex Twin-Width). A *convex uncontraction sequence of width d* of an ordered bipartite graph $G = (X, Y, E, \leq)$ is a sequence $\mathcal{P}_1, \dots, \mathcal{P}_m$ of divisions of $X \cup Y$ such that:

- $\mathcal{P}_1 = \{X, Y\}$;
- \mathcal{P}_m is a division into singletons;
- For $i = 1, \dots, m - 1$, the division \mathcal{P}_{i+1} is obtained from \mathcal{P}_i by splitting exactly one of the parts into two;
- For every part $U \in \mathcal{P}_i^X$, there are at most d parts $W \in \mathcal{P}_i^Y$ such that (U, W) is impure. For every part $W \in \mathcal{P}_i^Y$, there are at most d parts $U \in \mathcal{P}_i^X$ such that (U, W) is impure.

The *convex twin-width* $\text{ctww}(G)$ is the minimum d such that there is a convex uncontraction sequence of width d of G .

Lemma 4.11 ([GPT22], Lemmas 3.14 & 3.15). *For any ordered bipartite graph G , $\text{tww}(G) \leq \text{ctww}(G)$. For any bipartite graph $G = (X, Y, E)$, there is a total order \leq on $X \cup Y$ such that $\text{ctww}((X, Y, E, \leq)) \leq \text{tww}(G) + 1$.*

Definition 4.12 (Quasi-Chain Number). Let $G = (X, Y, E)$ be a bipartite graph. The *quasi-chain number* $\text{qch}(G)$ of G is the largest k for which there exist two sequences $x_1, \dots, x_k \in X$ and $y_1, \dots, y_k \in Y$ of not necessarily distinct vertices such that for each $i \in [k]$, one of the following holds:

1. x_i is adjacent to all of y_1, \dots, y_{i-1} and y_i is non-adjacent to all of x_1, \dots, x_{i-1} ; or,
2. x_i is non-adjacent to all of y_1, \dots, y_{i-1} and y_i is adjacent to all of x_1, \dots, x_{i-1} .

Lemma 4.13 ([GPT22], Lemma 3.3). *For every bipartite graph G ,*

$$\text{ch}(G) \leq \text{qch}(G) \leq 4 \cdot \text{ch}(G) + 4.$$

Definition 4.14 (Flip). A *flip* of a bipartite graph $G = (X, Y, E)$ is any graph $G' = (X, Y, E')$ obtained by choosing any $A \subseteq X, B \subseteq Y$ and negating the edge relation for every pair $(a, b) \in A \times B$.

For $q \in \mathbb{N}$ and a graph G , a graph G' is a *q -flip* of G if there is a sequence $G = G_0, G_1, \dots, G_r = G'$ such that $r \leq q$ and for each $i \in [r]$, G_i is a flip of G_{i-1} .

Lemma 4.15 ([GPT22], Main Lemma). *For all $k, d \in \mathbb{N}$, $k, d \geq 2$, there are $r, q \in \mathbb{N}$ satisfying the following. Let $G = (X, Y, E, \leq)$ be an ordered bipartite graph of convex twin-width at most d and quasi-chain number at most k . Then there is a division \mathcal{D} of G , sets $\mathcal{U}_1^X, \dots, \mathcal{U}_r^X \subseteq \mathcal{D}^X$, $\mathcal{U}_1^Y, \dots, \mathcal{U}_r^Y \subseteq \mathcal{D}^Y$, and a q -flip G' of G such that the following holds for $H := G'/\mathcal{D}$:*

1. For every edge (A, B) of H there exists $i \in [r]$ such that $A \in \mathcal{U}_i^X, B \in \mathcal{U}_i^Y$;
2. For each $i \in [r]$, the graph $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$ is a star forest. Moreover, for each star in $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$, with center C and leaves K_1, \dots, K_m , the quasi-chain number of $G[C \cup K_1 \cup \dots \cup K_m]$ is at most $k - 1$.

Lemma 4.16. *Let \mathcal{F} be any class of stable bipartite graphs with bounded twin-width. Then \mathcal{F} admits a constant-size equality-based labeling scheme.*

Proof. Since \mathcal{F} is stable, we have $\text{ch}(\mathcal{F}) = k$ and $\text{tw}(\mathcal{F}) = d$ for some constants k, d .

Decomposition tree. For any $G \in \mathcal{F}$, we construct a decomposition tree as follows. Let $k^* = \text{qch}(G)$ which satisfies $k^* \leq 4 \cdot \text{qch}(G) + 4$ by Lemma 4.13. The root of the tree is associated with G . Every node of the tree is associated with an induced subgraph G' of G , defined as follows:

For a node $G' = (X', Y', E')$ with $\text{qch}(G') \leq 1$, we have $\text{ch}(G') \leq \text{qch}(G') \leq 1$ by Lemma 4.13, so G' is P_4 .

For a node $G' = (X', Y', E')$ with $\text{qch}(G') = k' > 1$, let $G'_{\leq} = (X', Y', E', \leq)$ be an ordered bipartite graph with $\text{ctww}(G'_{\leq}) \leq d + 1$, which exists due to Lemma 4.11. Let \mathcal{D} be a division of G'_{\leq} , let $\mathcal{U}_1^X, \dots, \mathcal{U}_r^X \subseteq \mathcal{D}^X, \mathcal{U}_1^Y, \dots, \mathcal{U}_r^Y \subseteq \mathcal{D}^Y$, and let F be a q -flip of G'_{\leq} such that the following holds for $H = F/\mathcal{D}$:

1. For every edge (A, B) of H there exists $i \in [r]$ such that $A \in \mathcal{U}_i^X, B \in \mathcal{U}_i^Y$; and
2. For every $i \in [r]$, $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$ is a star forest such that for every star in this forest with center C and leaves K_1, \dots, K_m we have $\text{qch}(G'[C \cup K_1 \cup \dots \cup K_m]) \leq k' - 1$.

The above holds for some r, q determined by Lemma 4.15. Below we will let l^* be the maximum value of r over all nodes G' . Note that we may assume that each edge (A, B) of H appears in at most one set $\mathcal{U}_i := \mathcal{U}_i^X \cup \mathcal{U}_i^Y$. Otherwise we can remove the leaf incident with (A, B) from all but one set \mathcal{U}_i that induce a star forest containing (A, B) ; each star forest remains a star forest.

The child nodes of G' are the bipartite graphs $G'[C \cup K_1 \cup \dots \cup K_m]$ for each star (C, K_1, \dots, K_m) in the star forests induced by the sets \mathcal{U}_i . Observe that each child G'' has $\text{qch}(G'') < \text{qch}(G')$, so the depth of the tree is at most $\text{qch}(G) = k^* \leq 4k + 4$.

Labeling scheme. We will construct labels as follows. For a vertex x and a graph G containing x , we will write $L(x, G)$ for the label of x obtained by the following recursive labeling scheme. For each node G' of the decomposition tree, we assign labels to the vertices of G' inductively as follows.

1. If G' is a leaf node, so that $\text{qch}(G') \leq 1$, each x is assigned a label $L(x, G') = (0, p(x) \mid q(x))$, where $(p(x) \mid q(x))$ is the constant-size equality-based label in the graph G' , which is a P_4 -free bipartite graph. Recall that P_4 -free bipartite graphs are bipartite equivalence graphs, so there is a simple equality-based labeling scheme.
2. If G' is an inner node, perform the following. Let $\mathcal{U}_1^X, \mathcal{U}_1^Y, \dots, \mathcal{U}_r^X, \mathcal{U}_r^Y$ be the sets that induce star forests in H . For each i , fix an arbitrary numbering s_i to the stars in $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$.
 - (a) Let $G'_{\leq} = F_0, F_1, \dots, F_{q'} = F$ with $q' \leq q$ be the sequence of flips that take G'_{\leq} to F . To each vertex $x \in V(G')$ assign a binary vector $f(x) \in \{0, 1\}^q$ so that $f(x)_i = 1$ if and only if x is in the set that is flipped to get F_i from F_{i-1} .
 - (b) For each $i \in [r]$, any vertex $x \in V(G')$ belongs to at most one star $S_i = (C, K_1, \dots, K_m)$ in $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$. Append the tuple $(1, f(x) \mid s_1(S_1), \dots, s_r(S_r))$ where $s_i(S_i)$ is the number of the star $S_i = (C, K_1, \dots, K_m)$ in $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$ containing x . Then for each $i \in [r]$, append $L(x, G'_i)$ for $G'_i = G'[C \cup K_1 \cup \dots \cup K_m]$ induced by the star $S_i = (C, K_1, \dots, K_m)$.

Decoder. The decoder for this scheme is defined recursively as follows: Given labels $L(x, G), L(y, G)$ for vertices x, y :

1. If $L(x, G') = (0, p(x) | q(x))$, $L(y, G') = (0, p(y) | q(y))$ then output the adjacency of x, y in G' , which is a bipartite equivalence graph, as determined by the labels $(p(x) | q(x))$, $(p(y) | q(y))$.
2. If $L(x, G') = (1, f(x) | s_1(S_1), \dots, s_r(S_r))$, $L(y, G') = (1, f(y) | s_1(S'_1), \dots, s_r(S'_r))$ then let i be the unique value such that $s_i(S_i) = s_i(S'_i)$, if such a value exists. In this case, output the adjacency of x, y as determined from the labels $L(x, G'_i)$, $L(y, G'_i)$ where G'_i is the child corresponding to star $S_i = S'_i$. Otherwise, output the parity of

$$|\{i \in [q] : f(x)_i = f(y)_i = 1\}|.$$

Correctness. Let $x \in X, y \in Y$ be vertices of G .

Claim 4.17. *For any node $G' = (X', Y', E') \sqsubset G$ in the decomposition tree, there is at most one child $G'' \sqsubset G'$ such that $x, y \in V(G'')$.*

Proof of claim. Let F be the q -flip of G'_{\leq} , let \mathcal{D} the division of G'_{\leq} , and write $H = F/\mathcal{D}$. Let $A \subseteq X', B \subseteq Y'$ be the unique sets with $A, B \in \mathcal{D}$ such that $x \in A, y \in B$. Suppose that for some $i \in [r]$ there is a star $\{A, K_1, \dots, K_m\} \subseteq \mathcal{U}_i$ such that $B \in \{K_1, \dots, K_m\}$. Then (A, B) is an edge of H so, by assumption, (A, B) appears in exactly one star forest $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$. (A, B) also appears in exactly one star in the star forest $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$. So there is exactly one child $G'' = G'[C \cup K_1 \cup \dots \cup K_m]$ that contains both x and y . \square

It follows from the above claim that there is a unique maximal path $G = G_0, \dots, G_t$ in the decomposition tree, starting from the root, satisfying $x, y \in V(G_i)$ for each $i = 0, \dots, t$.

First we prove that the labeling scheme outputs the correct value on $G_t = (X', Y', E')$. If G_t is a leaf node then this follows from the labeling scheme for P_4 -free bipartite graphs. Suppose G_t is an inner node. Let $A \subseteq X', B \subseteq Y'$ be the unique sets $A, B \in V(H)$ such that $x \in A, y \in B$. Since G_0, \dots, G_t is a maximal path, it must be that there is no $i \in [r]$ and star S in $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$ such that both A and B are nodes of S . Then for every $i \in [r]$, let S_i, S'_i be the (unique) pair of stars in $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$ such that $A \in S_i, B \in S'_i$; since A, B are not nodes of the same star, we have $S_i \neq S'_i$ so $s_i(S_i) \neq s_i(S'_i)$ in the labels. So the decoder outputs the parity of

$$|\{i \in [q] : f(x)_i = f(y)_i = 1\}|.$$

We show that x, y are not adjacent in F . In this case, x, y are adjacent in G_t if and only if the pair (x, y) is flipped an odd number of times in the sequence $(G_t)_{\leq} = F_0, F_1, \dots, F_{q'} = F$; this is equivalent to there being an odd number of indices $i \in [q]$ such that $f(x)_i = f(y)_i$, so the decoder will be correct. For contradiction, assume that x, y are adjacent in F . Then by definition, (A, B) is an edge of H , so (A, B) must belong to some star S in some forest $H[\mathcal{U}_i^X, \mathcal{U}_i^Y]$. But then $s_i(S_i) = s_i(S'_i) = s_i(S)$, a contradiction. So x, y are not adjacent in F .

Now consider node G_i for $i = 0, \dots, t-1$. By definition, there is a child G_{i+1} that contains both x and y , so there exists $j \in [r]$ and a star $S = (C, K_1, \dots, K_m)$ in $H[\mathcal{U}_j^X, \mathcal{U}_j^Y]$ such that $x, y \in C \cup K_1 \cup \dots \cup K_m$. Then $s_j(S_j) = s_j(S'_j) = s_j(S)$ so the decoder will recurse on the child G_{i+1} . \square

4.1.4 First-Order Labeling Schemes & Distance Sketching

In this section we construct sketches for stable graph classes of bounded twin-width that replace the adjacency relation with a binary relation $\phi(x, y)$ on the vertices, that is defined by a formula ϕ in first-order logic. We will call such a sketch a ϕ -sketch.

First-order logic. A *relational vocabulary* τ is a set of relation symbols. Each relation symbol R has an *arity*, denoted $\text{arity}(R) \geq 1$. A *structure* \mathcal{A} of vocabulary τ , or τ -structure, consists of a set A , called the *domain*, and an interpretation $R^{\mathcal{A}} \subseteq A^{\text{arity}(R)}$ of each relation symbol $R \in \tau$. To briefly recall the syntax and semantics of first-order logic, we fix a countably infinite set of *variables*, for which we use small letters. *Atomic formulas of vocabulary* τ are of the form:

1. $x = y$ or
2. $R(x_1, \dots, x_r)$, meaning that $(x_1, \dots, x_r) \in R$,

where $R \in \tau$ is r -ary and x_1, \dots, x_r, x, y are variables. *First-order (FO) formulas* of vocabulary τ are inductively defined as either the atomic formulas, a Boolean combination $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, or a quantification $\exists x.\phi$ or $\forall x.\phi$, where ϕ and ψ are FO formulas. A *free variable* of a formula ϕ is a variable x with an occurrence in ϕ that is not in the scope of a quantifier binding x . We write $\phi(x_1, x_2, \dots, x_k)$ to show that the set of free variable of ϕ is $\{x_1, x_2, \dots, x_k\}$. By $\phi[t/x]$, we denote the formula that results from substituting t for free variable x in ϕ .

First-order labeling schemes. We fix a relational vocabulary τ that consists of a unique (symmetric) binary relational symbol E , and t unary relational symbols R_1, \dots, R_t . A τ -structure \mathcal{G} with domain V is a tuple $(V, E^{\mathcal{G}}, R_1^{\mathcal{G}}, \dots, R_t^{\mathcal{G}})$ where each $R_i^{\mathcal{G}} \subseteq V$ is an interpretation of the symbol R_i , and $E^{\mathcal{G}} \subseteq V \times V$ is an interpretation of the symbol E . For a graph class \mathcal{F} , we will write \mathcal{F}^τ for the set of τ -structures $\mathcal{G} = (V, E^{\mathcal{G}}, R_1^{\mathcal{G}}, \dots, R_t^{\mathcal{G}})$ where $(V, E^{\mathcal{G}})$ is a graph in \mathcal{F} . Let $\phi(x, y)$ be an FO formula of vocabulary τ . For a τ -structure $\mathcal{G} = (V, E^{\mathcal{G}}, R_1^{\mathcal{G}}, \dots, R_t^{\mathcal{G}})$ and $u, v \in V$, we write $\mathcal{G} \models \phi[u/x, v/y]$, if $\phi[u/x, v/y]$ is a true statement in \mathcal{G} . When the τ -structure is clear from context, we drop the superscript \mathcal{G} (and identify the relation symbol and its interpretation in \mathcal{G}).

Definition 4.18 (First-order sketches). For a symmetric first-order formula $\phi(x, y)$ with vocabulary τ , and a graph class \mathcal{F} , a ϕ -*sketch* with cost $c(n)$ and error δ is a pair of algorithms: a randomized *encoder* and a deterministic *decoder*. The encoder takes as input any τ -structure $G = (V, E, R_1, \dots, R_t) \in \mathcal{F}^\tau$ with $|V| = n$ vertices and outputs a (random) function $\text{sk} : V \rightarrow \{0, 1\}^{c(n)}$. The encoder and (deterministic) decoder $D : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}$ must satisfy the condition that for all $G \in \mathcal{F}^\tau$,

$$\forall u, v \in V(G) : \quad \mathbb{P}_{\text{sk}} \left[D(\text{sk}(u), \text{sk}(v)) = \mathbb{1}[G \models \phi[u/x, v/y]] \right] \geq 1 - \delta.$$

As usual, if left unspecified, we assume $\delta = 1/3$. We will write $\phi\text{-SK}(\mathcal{F})$ for the smallest function $c(n)$ such that there is a randomized ϕ -labeling scheme for \mathcal{F} with cost $c(n)$ and error $\delta = 1/3$. Setting $\delta = 0$ we obtain the notion of (*deterministic*) ϕ -*labeling scheme*.

Theorem 4.19. *Let \mathcal{F} be a stable, hereditary graph class with bounded twin-width, and let $\phi(x, y)$ be an FO formula of vocabulary τ . Then $\phi\text{-SK}(\mathcal{F}) = O(1)$.*

Proof. Given a τ -structure $G = (V, E, R_1, \dots, R_t) \in \mathcal{F}^\tau$, we denote by G^ϕ the graph with vertex set V and the edge set $E^\phi = \{(u, v) : G \models \phi[u/x, v/y]\}$. We also define $\mathcal{F}^\phi := \text{cl}(\{G^\phi : G \in \mathcal{F}^\tau\})$. By definition, \mathcal{F}^ϕ is hereditary. Furthermore, we note that \mathcal{F}^ϕ is an FO transduction of \mathcal{F} . Hence, [Theorem 4.4](#) implies that \mathcal{F}^ϕ is a stable class of bounded twin-width, and therefore by [Theorem 4.1](#), \mathcal{F}^ϕ admits a constant-size adjacency sketch. Any such sketch can be used as a ϕ -sketch for \mathcal{F} , since for any two vertices u, v in $G \in \mathcal{F}^\tau$ we have $G \models \phi[u/x, v/y]$ if and only if u and v are adjacent in G^ϕ . \square

As a corollary we obtain an answer to an open question of [\[Har20\]](#), who asked if the planar graphs admit constant-size sketches for $\text{dist}(x, y) \leq k$.

Corollary 4.20. *For any $k \in \mathbb{N}$, there is a constant-size sketch for planar graphs that decides $\text{dist}(x, y) \leq k$.*

Proof. The class of planar graphs has bounded twin-width [BKTW20] and bounded chain number (because it is of bounded arboricity). Hence, the result follows from Theorem 4.19 and the fact that the relation $\text{dist}(x, y) \leq k$ is expressible in first-order logic, e.g. via the following FO formula

$$\delta_k(x, y) := (\exists v_1, v_2, \dots, v_{k-1} : (E(x, v_1) \vee x = v_1) \wedge (E(v_1, v_2) \vee v_1 = v_2) \wedge \dots \wedge (E(v_{k-1}, y) \vee v_{k-1} = y)).$$

□

4.2 Graph Products

The Cartesian product of graphs is defined as follows.

Definition 4.21 (Cartesian Products). Let $d \in \mathbb{N}$ and G_1, \dots, G_d be any graphs. The Cartesian product $G_1 \square G_2 \square \dots \square G_d$ is the graph whose vertices are the tuples $(v_1, \dots, v_d) \in V(G_1) \times \dots \times V(G_d)$, and two vertices v, w are adjacent if and only if there is exactly one coordinate $i \in [d]$ such that $(v_i, w_i) \in E(G_i)$ and for all $j \neq i$, $v_j = w_j$. For any set of graphs \mathcal{F} , we will define the set of graphs \mathcal{F}^\square as all graphs obtained by taking a product of graphs in \mathcal{F} :

$$\mathcal{F}^\square := \{G_1 \square G_2 \square \dots \square G_d : d \in \mathbb{N}, \forall i \in [d] G_i \in \mathcal{F}\}.$$

We also write $G^{\square d} := G \square G \square \dots \square G$ for the d -wise product of G . For example, $P_2^{\square d}$ is the d -dimensional hypercube.

Although Cartesian products are extremely well-studied (e.g. see [CLR20] for results on universal graphs for subgraphs of products), it was not previously known whether the number of unique induced subgraphs of Cartesian products $G^{\square d}$ is at most $2^{O(n \log n)}$, let alone whether they admit adjacency labeling schemes of size $O(\log n)$. We resolve this question by proving that Cartesian products preserve constant-size PUGs and distance- k sketches. We prove the following theorem. Note that, unlike adjacency, distances are not necessarily preserved by taking induced subgraphs, so the distance- k result applies to \mathcal{F}^\square instead of its hereditary closure. The following theorem implies Theorem 1.2 from the introduction.

Theorem 4.22. *If \mathcal{F} is any class that admits a constant-size PUG (including any finite class), then $\text{cl}(\mathcal{F}^\square)$ admits a constant-size PUG. For any fixed k , if \mathcal{F} admits a constant-size distance- k sketch, then so does \mathcal{F}^\square .*

We obtain as a corollary the new result that $\text{cl}(\mathcal{F}^\square)$ is a factorial class with a $\text{poly}(n)$ -size universal graph, when \mathcal{F} is any class with a constant-size PUG (including any finite class), which follows by Proposition 3.5 and Proposition 2.5.

Corollary 4.23. *For any hereditary class \mathcal{F} that admits a constant-size PUG (in particular any finite class), the hereditary class $\text{cl}(\mathcal{F}^\square)$ has speed $2^{O(n \log n)}$ and admits a $\text{poly}(n)$ -size universal graph.*

Follow-up work [EHZ24] simplifies and extends our technique from this section to obtain optimal adjacency labels for *subgraphs* (not only induced subgraphs) of Cartesian products, improving upon the results of [CLR20].

Remark 4.24. There are 3 other common types of graph products [HIK11]: the strong product, the direct product, and the lexicographic product. Early versions of this paper showed that these graph products admit constant-cost communication protocols only in trivial cases; we omit this discussion from the current version.

We now prove [Theorem 4.22](#). The proof is illustrated in [Figure 3](#).

Proof of [Theorem 4.22](#). Let $s(k)$ be the size of the randomized distance k labeling scheme for \mathcal{F} , and let $D : \{0, 1\}^{s(k)} \times \{0, 1\}^{s(k)} \rightarrow \{0, 1\}$ be its decoder. We may assume that this scheme has error probability at most $\frac{1}{10k}$, at the expense of at most an $O(\log k)$ factor increase by [Proposition 3.4](#). For any $G = G_1 \square \cdots \square G_d$ where each $G_i \in \mathcal{F}$, we will construct a distance k labeling for G as follows.

Choose constants $m, t \in \mathbb{N}$ with $m \geq 9k^2$, $t \geq 9k$ and $mt \geq 27(k+1)^2$. Then:

1. For each $i \in [d]$, draw a random labeling $\ell_i : V(G_i) \rightarrow \{0, 1\}^{s(k)}$ from the distance k scheme for G_i .
2. For each $i \in [m]$, $j \in [t]$ and each vertex $x \in G$, initialize a vector $w^{(i,j)}(x) \in \{0, 1\}^{s(k)}$ to $\vec{0}$ and a bit $W^{(i,j)}(x)$ to 0.
3. For each $i \in [d]$ choose a uniformly random coordinate $b(i) \sim [m]$. For each $i \in [d]$ and each $v \in V(G_i)$ choose a random coordinate $c(i, v) \sim [t]$.
4. For each $x \in V(G)$, assign the label as follows. For each $i \in [d]$, update

$$\begin{aligned} w^{(b(i), c(i, x_i))}(x) &\leftarrow w^{(b(i), c(i, x_i))}(x) \oplus \ell_i(x_i) \\ W^{(b(i), c(i, x_i))}(x) &\leftarrow W^{(b(i), c(i, x_i))}(x) \oplus 1. \end{aligned}$$

Then the label for x is $w(x)$, the collection of all vectors $w^{(i,j)}(x)$ and bits $W^{(i,j)}(x)$ for $i \in [m]$, $j \in [t]$. So the size of the label is $mt \cdot (s(k) + 1)$.

The decoder is as follows. On receiving labels $w(x), w(y)$, perform the following:

1. For each $i \in [m], j \in [t]$, let $z^{(i,j)} = w^{(i,j)}(x) \oplus w^{(i,j)}(y)$ and $Z^{(i,j)} = W^{(i,j)}(x) \oplus W^{(i,j)}(y)$.
2. If there exists $i \in [m]$ for which the number of coordinates $j \in [t]$ such that $Z^{(i,j)} = 1$ is not in $\{0, 2\}$, output \perp .
3. If there are at least $2k + 1$ pairs $(i, j) \in [m] \times [t]$ such that $Z^{(i,j)} = 1$, output \perp .
4. Otherwise, there are $2q \leq 2k$ pairs

$$(i_1, (j_1)_1), (i_1, (j_1)_2), (i_2, (j_2)_1), (i_2, (j_2)_2), \dots, (i_q, (j_q)_1), (i_q, (j_q)_2) \in [m] \times [t]$$

such that $Z^{(i^*, j_1^*)} = Z^{(i^*, j_2^*)} = 1$ for each of these pairs $(i^*, j^*) \in \{(i_1, j_1), \dots, (i_q, j_q)\}$. For each $r \in [q]$, define

$$k_r = D(z^{(i_r, (j_r)_1)}, z^{(i_r, (j_r)_2)}) \in [k] \cup \{\perp\}.$$

If any of these values are \perp , output \perp . Otherwise output $\sum_{r=1}^q k_r$.

Claim 4.25. *Let $x, y \in G$ such that there are exactly $q \leq k$ coordinates $I \subseteq [d]$, $|I| = q$, such that $x_i \neq y_i$ for all $i \in I$. Then, with probability at least $2/3$, all of the following events occur:*

1. For all $i \in I$, either $\text{dist}(x_i, y_i) \leq k$ and $\text{dist}(x_i, y_i) = D(\ell_i(x_i), \ell_i(y_i))$, or $\text{dist}(x_i, y_i) > k$ and $D(\ell_i(x_i), \ell_i(y_i)) = \perp$;
2. The values $b(i)$, $i \in I$, are all distinct;
3. For all $i \in I$, $c(i, x_i) \neq c(i, y_i)$.

Proof of claim. By assumption, for any $i \in I$, the probability that $D(\ell_i(x_i), \ell_i(y_i))$ fails to output $\text{dist}(x_i, y_i)$ (if this is at most k) or \perp (if the distance is greater than k) is at most $1/9k$, so by the union bound, the probability that the first event fails to occur is at most $1/9$.

The probability that event 2 fails to occur is the probability that there exists a distinct pair $i, j \in I$ such that $b(i) = b(j)$; by the union bound, this is at most $|I|^2/m \leq k^2/m \leq 1/9$.

The probability that there exists $i \in I$ such that $c(i, x_i) = c(i, y_i)$ is at most $|I|/t \leq k/t \leq 1/9$. Therefore the probability that any one of these events fails to occur is at most $3/9 = 1/3$. \square

Claim 4.26. *For any $x, y \in G$ such that there are exactly $q \leq k$ coordinates $I \subseteq [d]$, $|I| = q$, such that $x_i \neq y_i$ for all $i \in I$. Then, if all 3 events in Claim 4.25 occur, the decoder outputs $\text{dist}(x, y)$ unless there is a coordinate j such that $\text{dist}(x_j, y_j) > k$, in which case it outputs \perp .*

Proof of claim. We first observe that the $2q$ pairs $P = \{(b(i), c(i, x_i)), (b(i), c(i, y_i))\}_{i \in I}$ are distinct, because each $b(i)$ is distinct and $c(i, x_i) \neq c(i, y_i)$. First, consider pairs $(b(i), c(i, x_i)), (b(i), c(i, y_i))$ for $i \in I$. Then, since $b(j) \neq b(i)$ for each $j \in I$ with $j \neq i$, and $c(i, y_i) \neq c(i, x_i)$, we have

$$\begin{aligned} Z^{(b(i), c(i, x_i))} &= W^{(b(i), c(i, x_i))}(x) \oplus W^{(b(i), c(i, x_i))}(y) \\ &= 1 \oplus \mathbf{1}[c(i, y_i) = c(i, x_i)] \oplus \left(\bigoplus_{j \neq i, b(j) = b(i)} \mathbf{1}[c(j, x_j) = c(i, x_i)] \right) \\ &\quad \oplus \left(\bigoplus_{j \neq i, b(j) = b(i)} \mathbf{1}[c(j, y_j) = c(i, x_i)] \right) \\ &= 1 \oplus \left(\bigoplus_{j \notin I, b(j) = b(i)} \mathbf{1}[c(j, x_j) = c(i, x_i)] \oplus \mathbf{1}[c(j, y_j) = c(i, x_i)] \right) \\ &= 1 \oplus \left(\bigoplus_{j \notin I, b(j) = b(i)} \mathbf{1}[c(j, x_j) = c(i, x_i)] \oplus \mathbf{1}[c(j, x_j) = c(i, x_i)] \right) \\ &= 1. \end{aligned}$$

Multiplying each indicator $\mathbf{1}[(b(j), c(j, x_j)) = (b(i), c(i, x_i))]$ by the vector $\ell_i(x_i)$, and each indicator $\mathbf{1}[(b(j), c(j, y_j)) = (b(i), c(i, x_i))]$ by the vector $\ell_i(y_i)$, we obtain the following:

$$z^{(b(i), c(i, x_i))} = w^{(b(i), c(i, x_i))} = \ell_i(x_i).$$

By similar reasoning, $Z^{(b(i), c(i, y_i))} = 1$, and

$$z^{(b(i), c(i, y_i))} = w^{(b(i), c(i, y_i))} = \ell_i(y_i).$$

Now consider any $i' \in [m]$, $j' \in [t]$ such that $(i', j') \notin P$. If there exists no $i \in [d]$, such that $i' = b(i)$, then clearly $W^{(i', j')}(x)$ and $W^{(i', j')}(y)$ remain 0 and $Z^{(i', j')} = 0$, so these entries do not contribute. Otherwise $i' = b(i)$ for some $i \in [d]$. If $c(i, x_i) \neq j' \neq c(i, y_i)$, again the label entries are not touched and $Z^{(i', j')} = 0$. Finally, assume $(i', j') = (b(i), c(i, x_i))$. Clearly $(b(j), c(j, x_j)) \neq (b(i), c(i, x_i))$ for

any $j \in I$ since $(i', j') \notin P$; in particular $i \notin I$. Thus $y_i = x_i$ by assumption. Then

$$\begin{aligned}
Z^{(b(i), c(i, x_i))} &= W^{(b(i), c(i, x_i))}(x) \oplus W^{(b(i), c(i, x_i))}(y) \\
&= \left(\bigoplus_{j: b(j)=b(i)} \mathbb{1}[c(j, x_j) = c(i, x_i)] \right) \oplus \left(\bigoplus_{j: b(j)=b(i)} \mathbb{1}[c(j, y_j) = c(i, x_i)] \right) \\
&= \left(\bigoplus_{j \notin I: b(j)=b(i)} \mathbb{1}[c(j, x_j) = c(i, x_i)] \oplus \mathbb{1}[c(j, y_j) = c(i, x_i)] \right) \\
&= \left(\bigoplus_{j \notin I: b(j)=b(i)} \mathbb{1}[c(j, x_j) = c(i, x_i)] \oplus \mathbb{1}[c(j, x_j) = c(i, x_i)] \right) = 0.
\end{aligned}$$

Since $x_i = y_i$, also $Z^{(b(i), c(i, y_i))} = 0$. We may then conclude that the $2q$ distinct pairs P are the only pairs (i, j) such that $Z^{(i, j)} = 1$. Therefore the decoder will output

$$\sum_{i \in I} D(w^{(b(i), c(i, x_i))}, w^{(b(i), c(i, y_i))}) = \sum_{i \in I} D(\ell_i(x_i), \ell_i(y_i)),$$

which is $\text{dist}(x, y)$ when each $\text{dist}(x_i, y_i) \leq k$ and \perp otherwise. \square

Suppose that $x, y \in V(G)$ have distance at most k . Then the above two claims suffice to show that the decoder will output $\text{dist}(x, y)$ with probability at least $2/3$.

Now suppose that $x, y \in V(G)$ have distance greater than k . There are three cases:

1. There are at least $k + 1$ coordinates $i \in [d]$ such that $x_i \neq y_i$. In this case, using [Proposition 4.27](#) below (with $\delta = \frac{1}{3}$, $u = mt$, $n = k + 1$, $k = k$) yields that vector Z has at least $k + 1$ 1-valued coordinates $Z^{(i, j)}$ with probability at least $2/3$, and the decoder will correctly output \perp .
2. There are at most k coordinates $i \in [d]$ such that $x_i \neq y_i$, and there is some coordinate i such that $\text{dist}(x_i, y_i) > k$. Then the above two claims suffice to show that the decoder will output \perp with probability at least $2/3$, as desired.
3. There are at most k coordinates $i \in [d]$ such that $x_i \neq y_i$, and all coordinates satisfy $\text{dist}(x_i, y_i) \leq k$. Then the above two claims suffices to show that the decoder will output $\text{dist}(x, y)$ with probability at least $2/3$.

This concludes the proof. \square

Proposition 4.27. *For any $0 < \delta < 1$ and any $u, k, n \in \mathbb{N}$, where $u \geq \frac{9(k+1)^2}{\delta}$ and $n > k$ the following holds. Write $e_i \in \mathbb{F}_2^u$ for the i th standard basis vector. Let $R_1, \dots, R_n \sim [u]$ be uniformly and independently random. Then $\mathbb{P}[|e_{R_1} + \dots + e_{R_n}| \leq k] < \delta$, where $|v|$ is the number of 1-valued coordinates in v and addition is in \mathbb{F}_2^u .*

Proof. Set $t = \lfloor \frac{1}{3}\sqrt{\delta u} \rfloor$. We first prove the following claim.

Claim. *For any vector $e \in \mathbb{F}_2^u$ and $S_1, \dots, S_t \sim [u]$, we have $\mathbb{P}[|e + e_{S_1} + \dots + e_{S_t}| \leq k] < \delta$.*

Proof. Observe that $t > \frac{1}{3}\sqrt{\delta u} - 1 \geq k$, and $t \leq \frac{1}{3}\sqrt{\delta u}$. Suppose first that $|e| > k + t$. Then $\mathbb{P}[|e + e_{S_1} + \dots + e_{S_t}| \leq k] = 0 < \delta$, as each of the vectors e_{S_i} can flip at most one coordinate in e from 1 to 0. So assume now that $|e| \leq k + t$. For a fixed $i \in [t]$ we have $\mathbb{P}[|e + e_{S_i}| < |e|] \leq \frac{k+t}{u} \leq \frac{2t}{u} \leq$

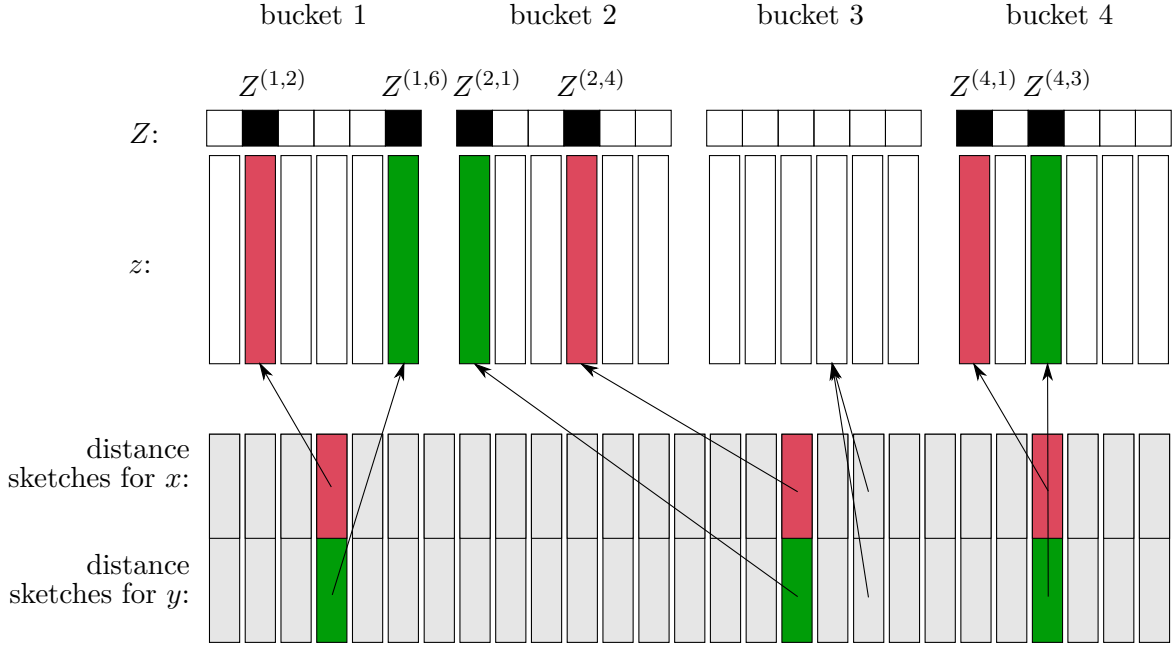


Figure 3: Small-distance sketch for Cartesian products. Along the bottom are the distances sketches for x_i (top) and y_i (bottom) for $i = 1$ to d . Where $x_i = y_i$, the sketches for x_i and y_i are equal and are colored grey; they cancel out in Z and z . Where $x_i \neq y_i$, the sketch for x_i is in red and the sketch for y_i is in green. For $i \neq j$ where $x_i \neq y_i$ and $x_j \neq y_j$, the sketches for x_i, y_i and x_j, y_j are mapped to different buckets with high probability (i.e., $b(i) \neq b(j)$) and the sketches for x_i and y_i are mapped to different locations in the same bucket with high probability (i.e., $c(i, x_i) \neq c(i, y_i)$).

$\frac{2\sqrt{\delta u}/3}{u} = \frac{2}{3} \frac{\sqrt{\delta}}{\sqrt{u}}$. Then for A , the event that $\exists i \in [t] : |e + e_{S_i}| < |e|$, we have $\mathbb{P}[A] \leq \frac{1}{3} \sqrt{\delta u} \cdot \frac{2}{3} \frac{\sqrt{\delta}}{\sqrt{u}} < \delta/2$.

Let further B be the event that $\exists i, j \in [t] : S_i = S_j$; then also $\mathbb{P}[B] \leq \binom{t}{2} \frac{1}{u} \leq \frac{t^2}{2u} < \delta/2$. If A and B both do *not* occur, we have $|e + e_{S_1} + \dots + e_{S_t}| = |e| + |e_{S_1}| + \dots + |e_{S_t}| \geq t > k$, so $\mathbb{P}[|e + e_{S_1} + \dots + e_{S_t}| \leq k] \leq \mathbb{P}[A \cup B] < \delta$, which proves the claim. \square

To prove the proposition we consider two cases. The first case is when $n \leq t$. In this case, as above, the probability that there are two distinct $i, j \in [n]$ such that $R_i = R_j$ is at most $\binom{n}{2} \frac{1}{u} \leq \binom{t}{2} \frac{1}{u} < \delta/2$. If all R_i are different then $|e_{R_1} + \dots + e_{R_n}| = n > k$. Hence, $\mathbb{P}[|e_{R_1} + \dots + e_{R_n}| \leq k] < \delta/2 < \delta$.

The second case is when $n > t$. In this case we apply the above claim with $e = e_{R_1} + \dots + e_{R_{n-t}}$ and $S_1 = R_{n-t+1}, \dots, S_t = R_n$. \square

5 Question II. Structure: Stability is Sometimes Sufficient

Recall Question II: What structures of a problem explain the existence or non-existence of a constant-cost protocol? Or equivalently, which structures of a hereditary graph class explain the existence or non-existence of a constant-size PUG?

In this section, we approach the question by looking at hereditary graph classes \mathcal{F} which satisfy the necessary condition of having factorial speed, but *do not* have a constant-size PUG, and asking: What structure of a subclass $\mathcal{H} \subset \mathcal{F}$ determines whether it *does* have a constant-size PUG? We hope to find statements of the form: “A subclass $\mathcal{H} \subset \mathcal{F}$ admits a constant-size PUG if and only if it has structure X ”.

We consider examples of two types of hereditary graph classes:

Section 5.1. Geometric intersection graphs are an important type of hereditary graph classes. These are classes graphs obtained by representing vertices as geometric objects (like lines in \mathbb{R}^2) and connecting vertices by an edge if their objects intersect. These types of graphs are important for both randomized communication complexity (e.g. [PS86, HHP⁺22, HHM23, HZ24]) and adjacency labeling schemes (e.g. [Spi03, Fit19]) and are the subject of open problems in both fields. Keeping in mind [Question 1.9](#), we consider two types of geometric intersection graphs – interval graphs and permutation graphs – which easily admit polynomial-size universal graphs, and ask what further structure is necessary and sufficient to get constant-size PUGs instead. We find that *stability* is the answer in both cases.

Section 5.2. Any hereditary subclass of bipartite graphs can be defined via a (possibly infinite) set \mathcal{H} of *bipartite* forbidden induced subgraphs. The ones obtained by finite sets \mathcal{H} are called *finitely-defined*, with the simplest example being the *monogenic* classes of bipartite graphs, obtained by forbidding exactly one bipartite graph H . It is known exactly for which graphs H the family of H -free bipartite graphs has the required condition of factorial speed, but it is not yet known whether all the H -free bipartite graph classes with factorial speed have polynomial-size universal graphs. We show that, for monogenic bipartite graph classes having factorial speed, stability is once again the necessary and sufficient condition to guarantee a constant-size PUG.

5.1 Interval & Permutation Graphs

Definition 5.1 (Interval graph). A graph G is an *interval graph* if there exists an *interval realization* $\ell, r : V(G) \rightarrow \mathbb{R}$ with $\ell(v) \leq r(v)$ for all $v \in V(G)$ so that $u, v \in V(G)$ are adjacent in G if and only if $[\ell(u), r(u)] \cap [\ell(v), r(v)] \neq \emptyset$.

The class of interval graphs admits a simple $O(\log n)$ -bit adjacency labeling scheme: Fix an interval realization of a given n -vertex interval graph G where all endpoints of the intervals are distinct integers in $[2n]$ and assign to each vertex a label consisting of the two endpoints of the corresponding interval.

Permutation graphs are another important class of geometric intersection graphs (like interval graphs, they are a subclass of segment intersection graphs). They also admit a straightforward $O(\log n)$ -bit adjacency labeling scheme that follows from their definition.

Definition 5.2 (Permutation Graph). A graph G is a *permutation graph* on n vertices if each vertex can be identified with a number $i \in [n]$, such that there is a permutation π of $[n]$ where i, j are adjacent if and only if $i < j$ and $\pi(i) > \pi(j)$.

In this section we prove [Theorem 1.4](#) from the introduction, restated more formally:

Theorem 5.3. *Let \mathcal{F} be any hereditary subclass of interval or permutation graphs. Then \mathcal{F} admits a constant-size PUG if and only if \mathcal{F} is stable.*

This proof requires new structural results for interval and permutation graphs⁶. We show that any stable subclass of interval graphs can be reduced to a class with bounded treewidth, and therefore has a constant-size PUG (implied by [Lemma 3.28](#)). A consequence of our proof is that

⁶A weaker version of our permutation graph result (without an explicit bound on the sketch size) also follows by our result for bounded twin-width combined with [BKTW20]. We keep our direct proof because: it gives an explicit bound on the sketch size; it is much clearer than the twin-width result; the decomposition scheme may be of independent interest; and it was proved independently of [GPT22], which we use for our twin-width result.

stable subclasses of interval graphs have bounded twin-width (which is not true for general interval graphs [BKTW20]). For permutation graphs, we give a new decomposition scheme whose depth can be controlled by the chain number.

We prove [Theorem 5.3](#) separately for interval graphs ([Theorem 5.8](#)) and for permutation graphs ([Theorem 5.21](#)) in the sections below.

5.1.1 Interval Graphs

The proof will rely on bounding the clique number of interval graphs with bounded chain number.

Lemma 5.4. *Let \mathcal{F} be a class of interval graphs with bounded clique number, i.e., there is a constant c so that for any graph $G \in \mathcal{F}$, the maximal clique size of G is at most c . Then \mathcal{F} admits a constant-size equality-based labeling scheme.*

Proof. Any interval graph is *chordal* and the treewidth of a chordal graph is one less its clique number. It follows that any interval graph G with clique number at most c has treewidth at most $c - 1$. Graphs of treewidth $c - 1$ have arboricity at most $O(c)$, and therefore, by [Lemma 3.28](#), \mathcal{F} admits a constant-size equality-based labeling scheme and an adjacency sketch of size $O(c)$. \square

It is not possible in general to bound the clique number of interval graphs with bounded chain number, because there may be an arbitrarily large set of vertices realized by identical intervals, which forms an arbitrarily large clique. Our first step is to observe that, for the purpose of designing an equality-based labeling scheme, we can ignore these duplicate vertices (called *true twins* in the literature).

Definition 5.5. For a graph $G = (V, E)$, two distinct vertices x, y are called *twins* if $N(x) \setminus \{y\} = N(y) \setminus \{x\}$, where $N(x), N(y)$ are the neighbourhoods of x and y in G . Twins x and y are *true twins* if they are adjacent, and *false twins* if they are not adjacent. The false-twin relation and true-twin relation are equivalence relations.

A graph is *true-twin-free* if it does not contain any vertices x, y that are true twins, and it is *false-twin-free* if it does not contain any vertices x, y that are false twins. It is *twin-free* if it is both true-twin-free and false-twin-free.

Lemma 5.6. *Let \mathcal{F} be any hereditary graph class and let \mathcal{F}' be either the set of true-twin-free members of \mathcal{F} , or the set of false-twin-free members of \mathcal{F} . If \mathcal{F}' admits an (s, k) -equality based labeling scheme, then \mathcal{F} admits an $(s, k + 1)$ -equality based labeling scheme.*

Proof. We prove the lemma for \mathcal{F}' being the true-twin-free members of \mathcal{F} ; the proof for the false-twin-free members is similar. Let $G \in \mathcal{F}$. We construct a true-twin-free graph $G' \in \mathcal{F}'$ as follows. Let $T_1, \dots, T_m \subseteq V(G)$ be the equivalence classes under the true-twin relation, so that for any i , any two vertices $x, y \in T_i$ are true twins. For each $i \in [m]$, let $t_i \in T_i$ be an arbitrary element, and let $T = \{t_1, \dots, t_m\}$. We claim that $G[T]$ is true-twin free.

Suppose for contradiction that t_i, t_j are true twins in $G[T]$. Let $x \in T_i, y \in T_j$. Since t_i, t_j are adjacent in $G[T]$, they are adjacent in G . Then x is adjacent to t_j since x, t_i are twins. Since t_j, y are twins, x is adjacent to y . So $G[T_i, T_j]$ is a biclique. Now suppose that $z \in T_k$ for some $k \notin \{i, j\}$, and assume z is adjacent to x . Then z is adjacent to t_i since x, t_i are twins, and t_i is adjacent to t_k since z, t_k are twins. Since t_i, t_j are twins, it also holds that t_j is adjacent to t_k and to z . So y is adjacent to z . Then for any z it holds that x, z are adjacent if and only if y, z are adjacent. So x, y are true twins, for any $x \in T_i, y \in T_j$. But then $T_i \cup T_j$ is an equivalence class under the true-twin relation, which is a contradiction.

Therefore, $G[T]$ is true-twin free and a member of \mathcal{F} , so $G[T] \in \mathcal{F}'$. For any $x \in V(G)$, assign the label $(p(t_i) \mid q(t_i), i)$ where $(p(t_i) \mid q(t_i))$ is the label of t_i in the equality-based labeling scheme for \mathcal{F}' , and $i \in [m]$ is the unique index such that $x \in T_i$. The decoder for \mathcal{F} performs the following on labels $(p(t_i) \mid q(t_i), i)$ and $(p(t_j) \mid q(t_j), j)$. If $i = j$ output 1; otherwise simulate the decoder for \mathcal{F}' on labels $(p(t_i) \mid q(t_j))$ and $(p(t_j) \mid q(t_j))$. For vertices x, y in G , if x, y are true twins then $i = j$ and the decoder outputs 1. Otherwise, the adjacency between x and y is equivalent to the adjacency between t_i, t_j in $G[T]$, which is computed by the decoder for \mathcal{F}' , as desired. \square

The true-twin-free interval graphs with bounded chain number also have bounded clique number.

Lemma 5.7. *Let G be a true-twin-free interval graph and let G contain a clique of c vertices. Then G has chain number at least $\lfloor \sqrt{c}/2 \rfloor$.*

Proof. Since G is interval, there is an interval realization $\ell, r : V(G) \rightarrow \mathbb{R}$ with $\ell(v) \leq r(v)$ for all $v \in V(G)$ so that $u, v \in V(G)$ are adjacent if and only if $[\ell(u), r(u)] \cap [\ell(v), r(v)] \neq \emptyset$. We can assume without loss of generality that no two endpoints are the same. We abbreviate $i(v) = [\ell(v), r(v)]$. Let $X = \{x_1, \dots, x_c\}$ be the c vertices that form a c -clique in G , arranged so that $\ell(x_1) < \ell(x_2) < \dots < \ell(x_c)$. Consider the sequence $r = (r(x_1), \dots, r(x_c))$ of right endpoints. By the Erdős-Szekeres theorem [ES35], any sequence of at least $R(k) = (k-1)^2 + 1$ distinct numbers contains either an increasing or a decreasing subsequence of length at least k . Setting $k = \lfloor \sqrt{c} \rfloor$, we have $R(k) \leq c$, so there is a clique over k vertices y_1, \dots, y_k with $\ell(y_1) < \dots < \ell(y_k)$ and either $r(y_1) < \dots < r(y_k)$ or $r(y_1) > \dots > r(y_k)$. Graphically speaking, the intervals for y_1, \dots, y_k either form a staircase or a (step) pyramid. In either case, $\ell(y_k) < \min\{r(y_1), r(y_k)\}$, so $m = (\min\{r(y_1), r(y_k)\} + \ell(y_k))/2$ is contained in all $i(y_j)$. We will assume the staircase case, $r(y_1) < \dots < r(y_k)$; see Figure 4. The pyramid case is similar.

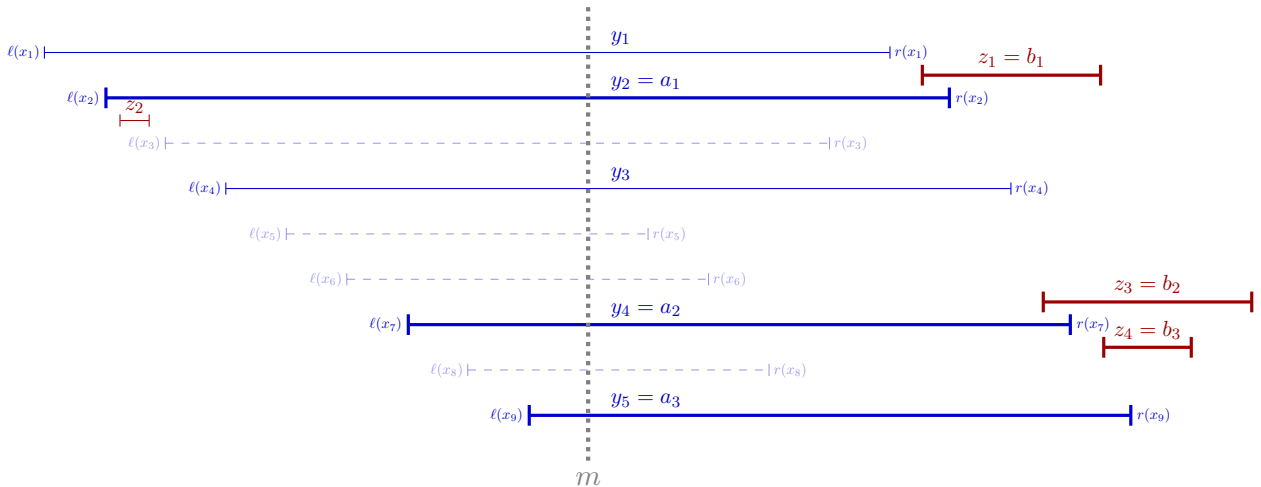


Figure 4: Example illustrating the notation from the proof of Lemma 5.7. The fat blue and red intervals, a_1, \dots, a_3 resp. b_1, \dots, b_3 , form an induced subgraph with chain number 3.

Now, since G is true-twin free, for every $v, v' \in X$, there must be $u \in V(G)$ with $i(v) \cap i(u) = \emptyset$ and $i(v') \cap i(u) \neq \emptyset$ or vice versa, so $i(u)$ must lie entirely to the left or entirely to the right of $i(v)$ or $i(v')$. In particular, for pair y_j, y_{j+1} with $j \in [k-1]$, there must be $z_j \in V(G)$ adjacent to exactly one of these vertices. So the endpoints of $i(z_j)$ are on the same side of m (“left” or “right”

of m) and the endpoint closer to m must be either between $\ell(y_j)$ and $\ell(y_{j+1})$ or between $r(y_j)$ and $r(y_{j+1})$.

Among the $k - 1$ intervals $i(z_1), \dots, i(z_{k-1})$, at least $h = \lceil (k - 1)/2 \rceil$ are on the same side of m . Assume the majority is on the right; the other case is similar. So for $1 \leq j_1 < \dots < j_h \leq k - 1$ intervals $i(z_{j_1}), \dots, i(z_{j_h})$ are all to the right of m . Define $B = (b_1, \dots, b_h) = (z_{j_1}, \dots, z_{j_h})$ and $A = (a_1, \dots, a_h) = (y_{j_1+1}, \dots, y_{j_h+1})$. By definition, $\ell(b_1) < r(a_1) < \ell(b_2) < r(a_2) < \dots < \ell(b_h) < r(a_h)$, so a_i is adjacent to b_j if and only if $j \leq i$. Hence $G[A, B]$ is isomorphic to H_h° . It is easy to check that $h = \lceil (\lfloor \sqrt{c} \rfloor - 1)/2 \rceil = \lfloor \sqrt{c}/2 \rfloor$. \square

With these preparations, the proof of the main result of this section becomes easy.

Theorem 5.8. *Let \mathcal{F} be a stable class of interval graphs. Then \mathcal{F} admits a constant-size equality-based adjacency labeling scheme, and hence $\text{SK}(\mathcal{F}_n) = O(1)$.*

Proof. Since \mathcal{F} is stable, we have $\text{ch}(\mathcal{F}) = k$ for some constant k . Let \mathcal{F}' be the set of true-twin-free members of \mathcal{F} , and let $G' \in \mathcal{F}'$. Then $\text{ch}(G') \leq k$, and hence the clique number of G' is at most $4(k + 1)^2$ by (contraposition of) Lemma 5.7. So \mathcal{F}' is a class of interval graphs with clique number bounded by $4(k + 1)^2$, and hence by Lemma 5.4, it admits a constant-size equality-based labeling scheme (and a size $O(k^2)$ adjacency sketch). By Lemma 5.6, so does \mathcal{F} . \square

Remark 5.9. We obtain an adjacency sketch of size $O(k^2)$ for interval graphs with chain number k . There is another, less direct proof of the above theorem that uses *twin-width* instead of Lemma 5.6, but does not recover this explicit bound on the sketch size. We prove in Section 4.1 that any stable class \mathcal{F} with bounded twin-width admits a constant-size equality-based labeling scheme. Although interval graphs do not have bounded twin-width, some subclasses of interval graphs (e.g. unit interval graphs) are known to have bounded twin-width [BKTW20]. Below we show that any stable class of interval graphs has bounded twin-width.

Proposition 5.10. *Let \mathcal{F} be a stable class of interval graphs. Then \mathcal{F} has bounded twin-width.*

Proof. It is known and can easily be deduced from the definition of twin-width, that if x and y are twins (true or false) in a graph G , then $\text{tw}(G) = \text{tw}(G - x)$, where $G - x$ is the graph obtained by removing x from G .

Let G be an arbitrary graph in \mathcal{F} and $\text{ch}(\mathcal{F}) = k$. Let G' be a twin-free induced subgraph of G that is obtained from G by iteratively removing twins. By the above, $\text{tw}(G) = \text{tw}(G')$, and, from Lemma 5.7, the clique number of G' is at most $4(k + 1)^2$. Then, by the argument in the proof of Lemma 5.4, G' has treewidth at most $4(k + 1)^2 - 1$, and therefore twin-width at most $2^{O(k^2)}$ [BD23]. Since G was chosen arbitrarily, the twin-width of any graph in \mathcal{F} is bounded by $2^{O(k^2)}$. \square

5.1.2 Permutation Graphs

We will denote by \prec the standard partial order on \mathbb{R}^2 , where $(x_1, x_2) \prec (y_1, y_2)$ if $x_1 \leq y_1$ and $x_2 \leq y_2$ and $(x_1, x_2) \neq (y_1, y_2)$.

The following alternative representation of permutation graphs is well-known (although one should note that adjacency is sometimes defined as between *incomparable* pairs, instead of comparable ones – this is equivalent since the complement of a permutation graph is again a permutation graph).

Proposition 5.11. *For any permutation graph G there is an injective mapping $\phi : V(G) \rightarrow \mathbb{R}^2$ such that $x, y \in V(G)$ are adjacent if and only if $\phi(x), \phi(y)$ are comparable in the partial order \prec . This mapping also satisfies the property that no two vertices x, y have $\phi(x)_i = \phi(y)_i$ for either $i \in [2]$.*

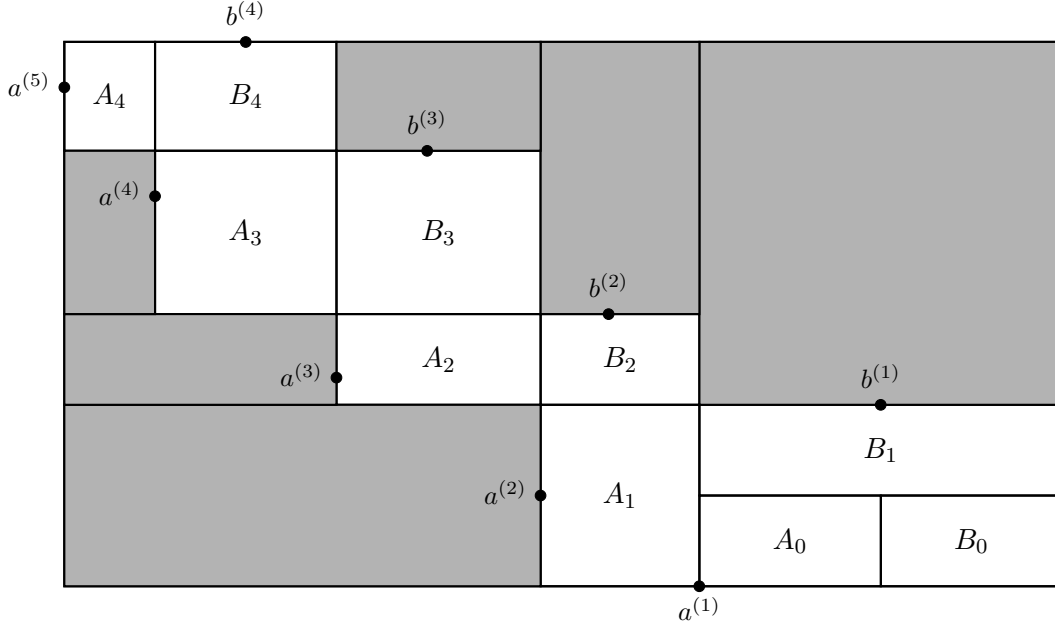


Figure 5: The permutation graph decomposition.

We will call this the \mathbb{R}^2 -representation of G . From now on we will identify vertices of G with their \mathbb{R}^2 -representation, so that a vertex x of G is a pair $(x_1, x_2) \in \mathbb{R}^2$. For a permutation graph G with fixed \mathbb{R}^2 -representation, any $i \in [2]$, and any $t_1 < t_2$, we define

$$V_i(t_1, t_2) := \{x \in V(G) : t_1 < x_i < t_2\}.$$

We need the following lemma, which gives a condition that allows us to increment the chain number.

Lemma 5.12. *For a graph G and a set $A \subset V(G)$, suppose $u, v \in V(G) \setminus A$ are vertices such that u has no neighbors in A , while v is adjacent to u and to every vertex in A . Then $\text{ch}(G[A \cup \{u, v\}]) > \text{ch}(G[A])$.*

Proof. Suppose $\text{ch}(G[A]) = k$ and let $\{a_1, \dots, a_k, b_1, \dots, b_k\}$ be the vertices such that a_i, b_j are adjacent if and only if $i \leq j$. Then set $a_{k+1} = u$ and $b_{k+1} = v$, and verify that vertices $\{a_1, \dots, a_{k+1}, b_1, \dots, b_{k+1}\}$ satisfy [Definition 3.10](#), so $\text{ch}(G[A \cup \{u, v\}]) \geq k + 1 > \text{ch}(G[A])$. \square

A bipartite graph $G = (X, Y, E)$ is called a *chain graph* if it is an induced subgraph of a half-graph. Chain graphs are exactly the $2K_2$ -free bipartite graphs, where $2K_2$ is the disjoint union of two edges.

Proposition 5.13. *For any $t \in \mathbb{R}$, any \mathbb{R}^2 -representation of a permutation graph G , and for each $i \in [2]$, $G[V_i(-\infty, t), V_i(t, \infty)]$ is a chain graph.*

Proof. Let $V_1(-\infty, t) = \{a^{(1)}, \dots, a^{(s)}\}$ and $V_1(t, \infty) = \{b^{(1)}, \dots, b^{(t)}\}$ where the vertices $\{a^{(i)}\}$ and $\{b^{(i)}\}$ are sorted in increasing order in the second coordinate. Since $a_1^{(i)} < t < b_1^{(j)}$ for every i, j , it holds that $a^{(i)}, b^{(j)}$ are adjacent if and only if $a_2^{(i)} \leq b_2^{(j)}$. To prove the statement we will show that $G[V_i(-\infty, t), V_i(t, \infty)]$ is $2K_2$ -free. Suppose, towards a contradiction, that $a^{(i_1)}, a^{(i_2)}, b^{(j_1)}, b^{(j_2)}$ induce a $2K_2$ in the graph, where $a^{(i_1)}$ is adjacent to $b^{(j_1)}$ and $a^{(i_2)}$ is adjacent to $b^{(j_2)}$. Assume,

without loss of generality, that $a_2^{(i_1)} < a_2^{(i_2)}$. Since $a^{(i_2)}$ is adjacent to $b^{(j_2)}$, we have that $a_2^{(i_2)} \leq b_2^{(j_2)}$, which together with the previous inequality imply that $a_2^{(i_1)} < b_2^{(j_2)}$, and hence $a^{(i_1)}$ is adjacent to $b^{(j_2)}$, a contradiction. \square

Any *proper* subclass of the class of chain graphs has a constant-size adjacency labeling scheme, because the class of chain graphs is a minimal factorial class (see [Appendix B](#)). We give an explicit bound on the size of the labeling scheme, so that we can get an explicit bound on the size of the labels for permutation graphs.

Proposition 5.14. *Let \mathcal{F} be a hereditary class of chain graphs of chain number at most k . Then \mathcal{F} admits an adjacency labeling scheme of size $O(\log k)$.*

Proof. Let $G \in \mathcal{F}$, so that $G \sqsubset H_r^{\circ\circ}$ for some $r \in \mathbb{N}$. Then we can partition $V(G)$ into independent sets A and B , such that the following holds. There is a total order \prec defined on $V(G) = A \cup B$ such that for $a \in A$ and $b \in B$, a, b are adjacent if and only if $a \prec b$. Then we may identify each $a \in A$ and each $b \in B$ with a number in $[n]$, such that the ordering \prec is the natural ordering on $[n]$.

Let $A_1, \dots, A_p \subseteq [n]$ be the set of (non-empty) maximal intervals such that each $A_i \subseteq A$, and let $B_1, \dots, B_q \subseteq [n]$ be the set of (non-empty) maximal intervals such that each $B_i \subseteq B$. We claim that $p, q \leq k + 1$. Suppose for contradiction that $p \geq k + 2$. Since A_1, \dots, A_p are maximal, there exist $b_1, \dots, b_{p-1} \in B$ such that $a_1 < b_1 < a_2 < b_2 < \dots < b_{p-1} < a_p$, where we choose $a_i \in A_i$ arbitrarily. But then $\{a_1, \dots, a_{p-1}\}, \{b_1, \dots, b_{p-1}\}$ is a witness that $\text{ch}(G) \geq p - 1 \geq k + 1$, a contradiction. A similar proof shows that $q \leq k + 1$.

We construct adjacency labels for G as follows. To each $x \in A$, assign 1 bit to indicate that $x \in A$, and append the unique number $i \in [k + 1]$ such that x belongs to interval A_i . To each $y \in B$, assign 1 bit to indicate that $y \in B$, and append the unique number $j \in [k + 1]$ such that $y \in B_j$. It holds that for $x \in A, y \in B$, x, y are adjacent if and only if $i \leq j$. Therefore, on seeing the labels for x and y , the decoder simply checks that $x \in A$ and $y \in B$ (or vice versa) and outputs 1 if $i \leq j$. \square

Definition 5.15 (Permutation Graph Decomposition; see [Figure 5](#)). For a permutation graph G with a fixed \mathbb{R}^2 -representation, where G, \overline{G} are both connected, we define the following partition. Let $a^{(1)}$ be the vertex with minimum $a_2^{(1)}$ coordinate, and let b be the vertex with maximum b_2 coordinate. If $b_1 < a_1^{(1)}$, perform the following. Starting at $i = 1$, construct the following sequence:

- (1) Let $b^{(i)}$ be the vertex with maximum $b_2^{(i)}$ coordinate among vertices with $b_1^{(i)} > a_1^{(i)}$.
- (2) For $i > 1$, let $a^{(i)}$ be the vertex with minimum $a_1^{(i)}$ coordinate among vertices with $a_2^{(i)} < b_2^{(i-1)}$.

Let β be the smallest number such that $b^{(\beta+1)} = b^{(\beta)}$ and α the smallest number such that $a^{(\alpha+1)} = a^{(\alpha)}$. Define these sets:

$$\text{For } 2 \leq i \leq \alpha, \text{ define } A_i := \{a^{(i+1)}\} \cup \left(V_1(a_1^{(i+1)}, a_1^{(i)}) \cap V_2(b_2^{(i-1)}, b_2^{(i)}) \right)$$

$$\text{For } 2 \leq i \leq \beta, \text{ define } B_i := \{b^{(i)}\} \cup \left(V_1(a_1^{(i)}, a_1^{(i-1)}) \cap V_2(b_2^{(i-1)}, b_2^{(i)}) \right)$$

$$A_1 := \{a^{(2)}\} \cup \left(V_1(a_1^{(2)}, a_1^{(1)}) \cap V_2(a_2^{(1)}, b_2^{(1)}) \right)$$

$$A_0 := \{a^{(1)}\} \cup \left(V_1(a_1^{(1)}, b_1^{(1)}) \cap V_2(a_2^{(1)}, a_2^{(2)}) \right)$$

$$B_1 := \{b^{(1)}\} \cup \left(V_1(a_1^{(1)}, \infty) \cap V_2(a_2^{(2)}, b_2^{(1)}) \right)$$

$$B_0 := \left(V_1(b_1^{(1)}, \infty) \cap V_2(a_2^{(1)}, a_2^{(2)}) \right).$$

If $b_1 > a_1^{(1)}$, define the map $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ as $\phi(x) = (-x_1, x_2)$ and apply ϕ to each vertex in the \mathbb{R}^2 -representation of G ; it is easily seen that the result is an \mathbb{R}^2 -representation of \overline{G} . Then apply the above process to \overline{G} .

It is necessary to ensure that $b^{(1)}$ is well-defined, i.e., that the set of points x with $x_1 > a_1^{(1)}$ is non-empty, so that the maximum is taken over a non-empty set.

Proposition 5.16. *If G is connected then there exists $x \in V(G)$ such that $x_1 > a_1^{(1)}$.*

Proof. Suppose otherwise. Then every $x \in V(G)$ distinct from $a^{(1)}$ has $x_2 > a_2^{(1)}$ by definition, and $x_1 < a_1^{(1)}$. But then x is not adjacent to $a^{(1)}$. So $a^{(1)}$ has no neighbors, contradicting the assumption that G is connected. \square

Proposition 5.17. *If G is connected and $b_1 < a_1^{(1)}$, then $b^{(1)} \neq b^{(2)}$.*

Proof. Suppose $b^{(2)} = b^{(1)}$. Then $b_2^{(1)} = b_2^{(2)}$ is maximum among all vertices x with $x_1 > a_1^{(2)}$, so $b_1 < a_1^{(2)}$. But all vertices x with $x_1 < a_1^{(2)}$ satisfy $x_2 > b_2^{(1)} = b_2^{(2)}$, so they cannot have an edge to $V_1(a_1^{(2)}, \infty)$. Both $V_1(a_1^{(2)}, \infty)$ and $V_1(-\infty, a_1^{(2)})$ are non-empty, so the graph is not connected. \square

Proposition 5.18. *If G is connected, the sets $\{A_i\}_{i=0}^\alpha, \{B_i\}_{i=0}^\beta$ form a partition of $V(G)$.*

Proof. Let $C = \{x : x_1 \geq a_1^{(\alpha)}, x_2 \leq b_2^{(\beta)}\}$. There are no vertices x with $x_1 > a_1^{(\alpha)}$ and $x_2 > b_2^{(\beta)}$, since this would contradict the definition of $b^{(\beta)}$; likewise, there are no vertices x with $x_1 < a_1^{(\alpha)}$ and $x_2 < b_2^{(\beta)}$, since this would contradict the definition of $a^{(\alpha)}$. Now suppose $x_1 < a_1^{(\alpha)}, x_2 > b_2^{(\beta)}$. Then x has no edge to any vertex $y \in C$. Then the set of vertices with $x_1 < a_1^{(\alpha)}, x_2 > b_2^{(\beta)}$ must be empty, otherwise $V(G)$ is partitioned into $C, V(G) \setminus C$ where $V(G) \setminus C \neq \emptyset$ has no edges to C .

Then we may assume that every vertex x is in C ; we will show that it belongs to some A_i or B_i . We may assume that x has distinct x_1, x_2 coordinates from all $a^{(i)}, b^{(i)}$, otherwise we would have $x = b^{(i)}$ or $x = a^{(i)}$, so x is an element of some A_i or B_i .

Let i be the smallest number such that $x_2 < b_2^{(i)}$. Suppose $i \geq 2$. By definition it must be that $x_1 > a_1^{(i+1)}$. If $x_1 > a_1^{(i-1)}$, then $x_2 < b_2^{(i-1)}$ by definition, which contradicts the choice of i . So it must be that $a_1^{(i+1)} < x_1 < a_1^{(i-1)}$ and $b_2^{(i-1)} < x_2 < b_2^{(i)}$. The set of points that satisfy this condition is contained in $A_i \cup B_i$. Now suppose $i = 1$. Again, it must be that $x_1 > a_1^{(2)}$ by definition, and also $a_2^{(1)} < x_2 < b_2^{(1)}$. The points satisfying these conditions are easily seen to be partitioned by A_0, B_0, A_1, B_1 . \square

For any subset $A \subset V(G)$ and any two vertices $u, v \in V(G) \setminus A$, we will say that u, v cover A if u has no edge into A and v is adjacent to u and every vertex in A . Then by [Lemma 5.12](#), if u, v cover A , then $\text{ch}(G) > \text{ch}(G[A])$.

Proposition 5.19. *If G is connected and $b_1 < a_1^{(1)}$, then for each $D \in \{A_i\}_{i=0}^\alpha \cup \{B_i\}_{i=0}^\beta$, $\text{ch}(G[D]) < \text{ch}(G)$.*

Proof. Each $x \in B_0$ satisfies $x_1 > b_1^{(1)} > a_1^{(1)}$ and $a_2^{(1)} < x_2 < a_2^{(2)} < b_2^{(1)}$, so $b^{(1)}$ has no neighbors in B_0 while $a^{(1)}$ is adjacent to $b^{(1)}$ and all vertices in B_0 , so $a^{(1)}, b^{(1)}$ cover B_0 .

Each $x \in B_1$ satisfies $x_1 > a_1^{(1)} > b_1^{(2)} > a_1^{(2)}$ and $a_2^{(1)} < x_2 < b_2^{(1)} < b^{(2)}$, so $b^{(2)}$ has no neighbors in B_1 and $a^{(2)}$ is adjacent to $b^{(2)}$ and all vertices in B_1 , so $a^{(2)}, b^{(2)}$ cover B_1 .

Each $x \in A_0$ satisfies $b_1^{(1)} > x_1 \geq a_1^{(1)} > a_1^{(2)}$ and $x_2 < a_2^{(2)} < b_2^{(1)}$, so $a^{(2)}$ has no neighbors in A_0 and $b^{(1)}$ is adjacent to $a^{(2)}$ and all vertices in A_0 , so $a^{(2)}, b^{(1)}$ cover A_0 .

Next we show that for any $1 \leq i \leq \alpha$, A_i is covered by $a^{(i)}, b^{(i)}$. By definition each $x \in A_i$ satisfies $x_1 < a_1^{(i)} < b_1^{(i)}$ and $a_2^{(i)} < b_2^{(i-1)} < x_2 < b_2^{(i)}$, so $a^{(i)}$ has no neighbors in A_i while $b^{(i)}$ is adjacent to $a^{(i)}$ and all vertices in A_i .

Finally, we show that for any $2 \leq i \leq \beta$, B_i is covered by $a^{(i)}, b^{(i-1)}$. By definition each $x \in B_i$ satisfies $a_1^{(i)} < x_1 < a_1^{(i-1)} < b_1^{(i-1)}$ and $a_2^{(i)} < b_2^{(i-1)} < x_2$, so $b^{(i-1)}$ has no neighbors in B_i while $a^{(i)}$ is adjacent to $b^{(i-1)}$ and all vertices in B_i . \square

Lemma 5.20. *Let $G \in \mathcal{P}$ be any permutation graph. Then one of the following holds:*

- (1) G is disconnected;
- (2) \overline{G} is disconnected;
- (3) There is a partition $V(G) = V_1 \cup \dots \cup V_m$ such that:
 - $\text{ch}(G[V_i]) < \text{ch}(G)$ for each $i \in [m]$, or $\text{ch}(\overline{G}[V_i]) < \text{ch}(\overline{G})$ for each $i \in [m]$;
 - For each $i \in [m]$, there is a set $J(i) \subset \{V_t\}_{t \in [m]}$ of at most 4 parts such that for each $W \in J(i)$, $G[V_i, W]$ is a chain graph; and
 - One of the following holds:
 - For all $i \in [m]$ and $W \in \{V_t\}_{t \in [m]} \setminus J(i)$, $G[V_i, W]$ is a co-biclique; or,
 - For all $i \in [m]$ and $W \in \{V_t\}_{t \in [m]} \setminus J(i)$, $G[V_i, W]$ is a biclique.

Proof. Assume G, \overline{G} are connected. Perform the decomposition of Definition 5.15. We will let $m = \alpha + \beta + 2$ and let V_1, \dots, V_m be the sets $\{A_i\}_{i=0}^\alpha \cup \{B_i\}_{i=0}^\beta$.

Case 1: $b_1 < a_1^{(1)}$. Then V_1, \dots, V_m is a partition due to Proposition 5.18, and $\text{ch}(G[V_i]) < \text{ch}(G), i \in [m]$ holds by Proposition 5.19. For $V_i = A_1$ we define the corresponding set $J(i) = \{A_0, B_0, B_1, B_2\}$. Since all sets V_i, V_j with $i \neq j$ are separated by a horizontal line or a vertical line, it holds by Proposition 5.13 that $G[V_i, V_j]$ is a chain graph. Now let $W \notin J(i)$. Observe that all $x \in W$ must satisfy $x_1 < a_1^{(2)}$ and $x_2 > b_2^{(1)}$, so x is not adjacent to any vertex in A_1 . So $G[A_1, W]$ is a co-biclique.

Now for $V_i \in \{A_0, B_0, B_1\}$, we let $J(i) = \{A_0, B_0, A_1, B_1\} \setminus \{V_i\}$. Similar arguments as above hold in this case to show that $G[V_i, W]$ is a co-biclique for each $W \notin J(i)$.

For $V_i = A_j$ for some $j > 1$, we define $J(i) = \{B_j, B_{j+1}\}$. For any $W \notin J(i)$ with $W \neq A_j$, it holds either that all $x \in W$ satisfy $x_1 < a_1^{(i+1)}$ and $x_2 > b_2^{(j)}$, or that all $x \in W$ satisfy $x_1 \geq a_1^{(j)}$ and $x_2 \leq b_2^{(j-1)}$; in either case x is not adjacent to any vertex in A_j , so $G[A_j, W]$ is a co-biclique.

For $V_i = B_j$ for some $j > 1$, we define $J(i) = \{A_j, A_{j-1}\}$. Similar arguments to the previous case show that $G[B_j, W]$ is a co-biclique for each $W \notin J(i), W \neq B_j$. This concludes the proof for case 1.

Case 2: $b_1 > a_1^{(1)}$. In this case we transform the \mathbb{R}^2 -representation of G using ϕ to obtain an \mathbb{R}^2 -representation of \overline{G} and apply the arguments above to obtain V_1, \dots, V_m such that $\text{ch}(\overline{G}[V_i]) < \text{ch}(\overline{G})$ for each $i \in [m]$, and each $V_j \in \{V_t\}_{t \in [m]} \setminus J(i)$ satisfies that $\overline{G}[V_i, V_j]$ is a co-biclique; then $G[V_i, V_j]$ is a biclique as desired. \square

Theorem 5.21. *Let \mathcal{F} be a stable subclass of permutation graphs. Then \mathcal{F} admits a constant-size equality-based labeling scheme, and hence $\text{SK}(\mathcal{F}) = O(1)$.*

Proof. Since \mathcal{F} is stable, we have $\text{ch}(\mathcal{F}) = k$ for some constant k .

We apply an argument similar to Lemma 5.25. For any $G \in \mathcal{F}$, we construct a decomposition tree where each node is associated with either an induced subgraph of G , or a bipartite induced

subgraph of G , with the root node being G itself. For each node G' , we decompose G' into children as follows,

1. If G' is a chain graph, the node is a leaf node.
2. If G' is disconnected, call the current node a D -node, and let the children G_1, \dots, G_t be the connected components of G' .
3. If $\overline{G'}$ is disconnected, call the current node a \overline{D} -node, and let $C_1, \dots, C_t \subseteq V(G')$ be such that $\overline{G'}[C_i], i \in [t]$ are the connected components of $\overline{G'}$. Define the children to be $G_i = G[C_i], i \in [t]$.
4. Otherwise construct V_1, \dots, V_m as in [Lemma 5.20](#) and let the children be $G[V_i]$ for each $i \in [m]$ and $G[V_i, V_j]$ for each i, j such that $i \in [m]$ and $V_j \in J(i)$. Call this node a P -node.

We will show that this decomposition tree has bounded depth. As in the decomposition for bipartite graphs, on any leaf-to-root path there cannot be two adjacent D -nodes or \overline{D} -nodes. As in the proof of [Claim 5.48](#), if G'' is associated with a D -node and its parent G' is associated with a \overline{D} -node, and G''' is any child of G'' , then $\text{ch}(G') > \text{ch}(G''')$. On the other hand, if G'' is associated with a \overline{D} -node and its parent is associated with a D -node, then $\text{ch}(\overline{G'}) > \text{ch}(\overline{G''})$.

Now consider any P -node associated with G' , with child G'' . By [Lemma 5.20](#), it holds that either G'' is a bipartite induced subgraph of G' that is a chain graph, or G'' has $\text{ch}(G'') < \text{ch}(G')$ or $\text{ch}(\overline{G''}) < \text{ch}(\overline{G'})$. It is easy to verify that $\text{ch}(\overline{G}) \leq \text{ch}(G) + 1$ for any graph G . Now, since every sequence G''', G'', G' of inner nodes along the leaf-to-root path in the decomposition tree must satisfy $\text{ch}(G''') < \text{ch}(G')$ or $\text{ch}(\overline{G''}) < \text{ch}(\overline{G'})$ and $\text{ch}(\overline{G}) \leq k + 1$, it must be that the depth of the decomposition tree is at most $2(2k + 1)$.

Now we construct an equality-based labeling scheme. For a vertex x , we construct a label at each node G' inductively as follows.

1. If G' is a leaf node, it is a chain graph with chain number at most k . We may assign a label of size $O(\log k)$ due to [Proposition 5.14](#).
2. If G' is a D -node with children G_1, \dots, G_t , append the pair $(D | i)$ where the equality code i is the index of the child G_i that contains x , and recurse on G_i .
3. If G' is a \overline{D} -node with children G_1, \dots, G_t , append the pair $(\overline{D} | i)$ where the equality code i is the index of the child G_i that contains x , and recurse on G_i .
4. If G' is a P -node, let V_1, \dots, V_m be partition of $V(G')$ as in [Lemma 5.20](#), and for each i let $J(i)$ be the (at most 4) indices such that $G'[V_i, V_j]$ is a chain graph when $j \in J(i)$. Append the tuple

$$(P, b, \ell_1(x), \ell_2(x), \ell_3(x), \ell_4(x) | i, j_1, j_2, j_3, j_4)$$

where b indicates whether all $G'[V_i, V_j], j \notin J(i)$ are bicliques or co-bicliques; the equality code i is the index such that $x \in V_i$, the equality codes j_1, \dots, j_4 are the elements of $J(i)$, and $\ell_s(x)$ is the $O(\log k)$ -bits adjacency label for x in the chain graph $G'[V_i, V_{j_s}]$. Then, recurse on the child $G'[V_i]$.

Given labels for x and y , which are sequences of the tuples above, the decoder iterates through the pairs and performs the following. On pairs $(D, i), (D, j)$ the decoder outputs 0 if $i \neq j$, otherwise it continues. On pairs $(\overline{D}, i), (\overline{D}, j)$, the decoder outputs 1 if $i \neq j$, otherwise it continues. On tuples

$$\begin{aligned} &(P, b, \ell_1(x), \ell_2(x), \ell_3(x), \ell_4(x) | i, j_1, j_2, j_3, j_4) \\ &(P, b, \ell_1(y), \ell_2(y), \ell_3(y), \ell_4(y) | i', j'_1, j'_2, j'_3, j'_4), \end{aligned}$$

the decoder continues to the next tuple if $i = i'$. Otherwise, the decoder outputs 1 if $i \notin \{j'_1, \dots, j'_4\}$ and $i' \notin \{j_1, \dots, j_4\}$ and b indicates that $G'[V_i, V_j]$ are bicliques for $j \notin J(i)$; it outputs 0 if b indicates otherwise. If $i = j'_s$ and $i' = j_t$ then the decoder outputs the adjacency of x, y using the labels $\ell_t(x), \ell_s(y)$. On any tuple that does not match any of the above patterns, the decoder outputs 0.

Since the decomposition tree has depth at most $2(2k + 1)$, each label consists of $O(k)$ tuples. Each tuple contains at most $O(\log k)$ prefix bits (since adjacency labels for the chain graph with chain number at most k have size at most $O(\log k)$) and at most 5 equality codes. So this is an $(O(k \log k), O(k))$ -equality-based labeling scheme.

The correctness of the labeling scheme follows from the fact that at any node G' , if x, y belong to the same child of G' , the decoder will continue to the next tuple. If G' is the lowest common ancestor of x, y in the decomposition tree, then x and y are adjacent in G if and only if they are adjacent in G' . If G' is a D - or \overline{D} -node then adjacency is determined by the equality of i, j in the tuples $(D | i), (D | j)$ or $(\overline{D} | i), (\overline{D} | j)$. If G' is a P -node and $i \notin J(i')$ (equivalently, $i' \notin J(i)$) then adjacency is determined by b . If $i \in J(i')$ (equivalently, $i' \in J(i)$) then $i = j'_s$ and $i' = j_t$ for some s, t , and the adjacency of x, y is equivalent to their adjacency in $G[V_i, V_{i'}] = G[V_{j'_s}, V_{j_t}]$, which is a chain graph, and it is determined by the labels $\ell_t(x), \ell_s(y)$. \square

Remark 5.22. We get an explicit $O(k \log k)$ bound on the size of the adjacency sketch in terms of the chain number k , due to [Proposition 3.23](#); this explicit bound would not arise from the alternate proof that goes through the twin-width (proper subclasses of permutation graphs have bounded twin-width [[BKTW20](#)], so we could apply [Theorem 4.1](#)).

5.2 Monogenic Bipartite Graphs

As explained in [Section 3.2](#), any constant-cost communication problem is equivalent to the problem of deciding adjacency in a hereditary class of *bipartite* graphs. In this section we answer [Question 1.8](#) for the hereditary classes of bipartite graphs which have at most factorial speed and are defined by a single forbidden induced bipartite subgraph. The following theorem is a formal restatement of [Theorem 1.5](#) from the introduction:

Theorem 5.23. *Let H be a bipartite graph such that the class of H -free bipartite graphs is factorial. Then any hereditary subclass \mathcal{F} of the H -free bipartite graphs has a constant-size PUG if and only if \mathcal{F} is stable.*

To prove this theorem, we require new structural results for some classes of bipartite graphs. Previous work [[All09](#), [LZ17](#)] has shown that a class of H -free bipartite graphs is factorial only when H is an induced subgraph of $P_7, S_{1,2,3}$, or one of the infinite set $\{F_{p,q}^*\}_{p,q \in \mathbb{N}}$ (defined in [Section 5.2.2](#)). We construct a new decomposition scheme for the $F_{p,q}^*$ -free graphs whose depth is controlled by the chain number, and we show that the chain number controls the depth of the decomposition scheme from [[LZ17](#)] for P_7 -free graphs.

As a result, we get a poly(n)-size universal graph for any stable subclass of the P_7 -free bipartite graphs. The P_7 -free bipartite graphs form a factorial class, but existence of a poly(n)-size universal graph for this class is not known. We take this as evidence that, independent of randomized communication, the study of stable graph classes might allow progress on the IGQ.

We remark that, under a conjecture of [[LZ17](#)], if our theorem was proved for the classes of bipartite graphs obtained by excluding only *two* graphs H_1, H_2 , it would establish that stability is characteristic of the constant-PUG factorial classes of bipartite graphs that are obtained by excluding any *finite* set of induced subgraphs.

5.2.1 Decomposition Scheme for Bipartite Graphs

In this section we define a decomposition scheme for bipartite graphs that we will use to establish constant-size adjacency sketches for the stable subclasses of factorial monogenic classes of bipartite graph.

Definition 5.24 ((\mathcal{Q}, k) -decomposition tree). Let $G = (X, Y, E)$ be a bipartite graph, $k \geq 2$, and let \mathcal{Q} be a hereditary class of bipartite graphs. A graph G admits a (\mathcal{Q}, k) -decomposition tree of depth d if there is a tree of depth d of the following form, with G as the root. Each node of the tree is a bipartite graph $G' = G[X', Y']$ for some $X' \subseteq X, Y' \subseteq Y$, labelled with either L, D, \overline{D} , or P as follows

- (1) L (*leaf node*): The graph G' belongs to \mathcal{Q} .
- (2) D (D -node): The graph G' is disconnected. There are sets $X'_1, \dots, X'_t \subseteq X'$ and $Y'_1, \dots, Y'_t \subseteq Y'$ such that $G[X'_1, Y'_1], \dots, G[X'_t, Y'_t]$ are the connected components of G' . The children of this decomposition tree node are $G[X'_1, Y'_1], \dots, G[X'_t, Y'_t]$.
- (3) \overline{D} (\overline{D} -node): The graph $\overline{G'}$ is disconnected. There are sets $X'_1, \dots, X'_t \subseteq X'$ and $Y'_1, \dots, Y'_t \subseteq Y'$ such that $\overline{G[X'_1, Y'_1]}, \dots, \overline{G[X'_t, Y'_t]}$ are the connected components of $\overline{G'}$. The children of this decomposition tree node are $G[X'_1, Y'_1], \dots, G[X'_t, Y'_t]$.
- (4) P (P -node): The vertex set of G' is partitioned into at most $2k$ non-empty sets $X'_1, X'_2, \dots, X'_p \subseteq X'$ and $Y'_1, Y'_2, \dots, Y'_q \subseteq Y'$, where $p \leq k, q \leq k$. The children of this decomposition tree node are $G[X'_i, Y'_j]$, for all $i \in [p], j \in [q]$. We say that the P -node G' is *specified* by the partitions X'_1, X'_2, \dots, X'_p and Y'_1, Y'_2, \dots, Y'_q .

Lemma 5.25. *Let $k \geq 2$ and $d \geq 1$ be natural constants, and let \mathcal{Q} be a class of bipartite graphs that admits a constant-size equality-based adjacency labeling scheme. Let \mathcal{F} be a class of bipartite graphs such that each $G \in \mathcal{F}$ admits a (\mathcal{Q}, k) -decomposition tree of depth at most d . Then \mathcal{F} admits a constant-size equality-based adjacency labeling scheme.*

Proof. Let $G = (X, Y, E) \in \mathcal{F}$. We fix a (\mathcal{Q}, k) -decomposition tree of depth at most d for G . For each node v in the decomposition tree we write G_v for the induced subgraph of G associated with node v . Each leaf node v has $G_v \in \mathcal{Q}$. For some constants s and r , we fix an (s, r) -equality-based adjacency labeling scheme for \mathcal{Q} , and for each leaf node v , we denote by ℓ'_v the function that assigns labels to the vertices of G_v under this scheme.

For each vertex x we will construct a label $\ell(x)$ that consists of a constant number of tuples (as in Remark 3.22), where each tuple contains one prefix of at most two bits, and at most two equality codes. First, we add to $\ell(x)$ a tuple $(\alpha(x) \mid -)$, where $\alpha(x) = 0$ if $x \in X$, and $\alpha(x) = 1$ if $x \in Y$. Then we append to $\ell(x)$ tuples defined inductively. Starting at the root of the decomposition tree, for each node v of the tree where G_v contains x , we add tuples $\ell_v(x)$ defined as follows. Write $X' \subseteq X, Y' \subseteq Y$ for the vertices of G_v .

- If v is a leaf node, then $G_v \in \mathcal{Q}$, and we define $\ell_v(x) = (L \mid -), \ell'_v(x)$.
- If v is a D -node then G_v is disconnected, with sets $X'_1, \dots, X'_t \subseteq X', Y'_1, \dots, Y'_t \subseteq Y'$ such that the children v_1, \dots, v_t are the connected components $G_v[X'_1, Y'_1], \dots, G_v[X'_t, Y'_t]$ of G_v . We define $\ell_v(x) = (D \mid j), \ell_{v_j}(x)$, where $j \in [t]$ is the unique index such that x belongs to the connected component $G_v[X'_j, Y'_j]$, and $\ell_{v_j}(x)$ is the inductively defined label for the child node v_j .

- If v is a \overline{D} -node then \overline{G}_v is disconnected, with sets $X'_1, \dots, X'_t \subseteq X', Y'_1, \dots, Y'_t \subseteq Y'$ such that $\overline{G}_v[X'_1, Y'_1], \dots, \overline{G}_v[X'_t, Y'_t]$ are the connected components of \overline{G}_v , and the children v_1, \dots, v_t of v are the graphs $G_v[X'_1, Y'_1], \dots, G_v[X'_t, Y'_t]$. We define $\ell_v(x) = (\overline{D} | j), \ell_{v_j}(x)$, where $j \in [t]$ is the unique index such that x belongs to $G_v[X'_j, Y'_j]$, and $\ell_{v_j}(x)$ is the inductively defined label for the child node v_j .
- If v is a P -node then let $X'_1, \dots, X'_p \subseteq X', Y'_1, \dots, Y'_q \subseteq Y'$ be the partitions of X', Y' with $p, q \leq k$. For each $(i, j) \in [p] \times [q]$, let $v_{i,j}$ be the child node of v corresponding to the subgraph $G_v[X'_i, Y'_j]$. If $x \in X$, then there is a unique $i \in [p]$ such that $x \in X'_i$, and we define $\ell_v(x) = (P | i, q), \ell_{v_{i,1}}(x), \dots, \ell_{v_{i,q}}(x)$, where $\ell_{v_{i,j}}(x)$ is the label assigned to x at node $v_{i,j}$. If $x \in Y$, then we define $\ell_v(x) = (P | i, p), \ell_{v_{1,i}}(x), \dots, \ell_{v_{p,i}}(x)$, where $i \in [q]$ is the unique index such that $x \in Y'_i$.

First, we will estimate the size of the label $\ell(x)$ produced by the above procedure. For every leaf node v , the label $\ell_v(x)$ of x is a tuple consisting of an s -bit prefix and r equality codes. Let $f(i)$ be the maximum number of tuples added to $\ell(x)$ by a node v at level i of the decomposition tree, where the root node belongs to level 0. Then, by construction, $f(i) \leq 1 + k \cdot f(i+1)$ and $f(d-1) = 1$, which implies that the total number of tuples in $\ell(x)$ does not exceed $f(0) \leq k^d$. Since every tuple contains a prefix with at most $s' = \max\{2, s\}$ bits, and at most $r' = \max\{2, r\}$ equality codes, we have that the label $\ell(x)$ contains a prefix with at most $s'k^d$ bits, and at most $r'k^d$ equality codes.

We will now show how to use the labels to define an equality-based adjacency decoder. Let x and y be two arbitrary vertices of G . The decoder first checks the first tuples $(\alpha(x) | -)$ and $(\alpha(y) | -)$ of the labels $\ell(x)$ and $\ell(y)$ respectively, to ensure that x, y are in different parts of G and outputs 0 if they are not. We may now assume $x \in X, y \in Y$. The remainder of the labels are of the form $\ell_v(x)$ and $\ell_v(y)$, where v is the root of the decomposition tree.

- If the labels $\ell_v(x), \ell_v(y)$ are of the form $(L | -), \ell'_v(x)$ and $(L | -), \ell'_v(y)$, then the decoder simulates the decoder for the labeling scheme for \mathcal{Q} , on inputs $\ell'_v(x), \ell'_v(y)$, and outputs the correct adjacency value.
- If the labels $\ell_v(x), \ell_v(y)$ are of the form $(D | i), \ell_{v_i}(x)$ and $(D | j), \ell_{v_j}(y)$, the decoder outputs 0 when $i \neq j$ (i.e., x, y are in different connected components of G_v), and otherwise it recurses on $\ell_{v_i}(x), \ell_{v_i}(y)$.
- If the labels $\ell_v(x), \ell_v(y)$ are of the form $(\overline{D} | i), \ell_{v_i}(x)$ and $(\overline{D} | j), \ell_{v_j}(y)$, the decoder outputs 1 when $i \neq j$ (i.e., x, y are in different connected components of \overline{G}_v and therefore they are adjacent in G_v), and otherwise it recurses on $\ell_{v_i}(x), \ell_{v_i}(y)$.
- If the labels $\ell_v(x), \ell_v(y)$ are of the form $(P | i, q), \ell_{v_{i,1}}(x), \dots, \ell_{v_{i,q}}(x)$ and $(P | j, p), \ell_{v_{1,j}}(y), \dots, \ell_{v_{p,j}}(y)$ the decoder recurses on $\ell_{v_{i,j}}(x)$ and $\ell_{v_{i,j}}(y)$.

It is routine to verify that the decoder will output the correct adjacency value for x, y . □

Remark 5.26 ((\mathcal{Q}, k) -tree for general graphs). A similar decomposition scheme can be used for non-bipartite graph classes; we do this for permutation graphs in [Section 5.1.2](#).

5.2.2 Monogenic Classes of Bipartite Graphs

Let \mathcal{H} be a *finite* set of bipartite graphs. It is known [[All09](#)] that if the class of \mathcal{H} -free bipartite graphs is at most factorial, then \mathcal{H} contains a forest and a graph whose bipartite complement is a forest. The converse was conjectured in [[LZ17](#)], where it was verified for monogenic classes of

bipartite graphs. More specifically, it was shown that, for a colored bipartite graph H , the class of H -free bipartite graphs is at most factorial if and only if both H and its bipartite complement is a forest. It is not hard to show that a colored bipartite graph H is a forest and its bipartite complement is a forest if and only if H is an induced subgraph of $S_{1,2,3}$, P_7 , or one of the graphs $F_{p,q}^*$, $p, q \in \mathbb{N}$ defined below.

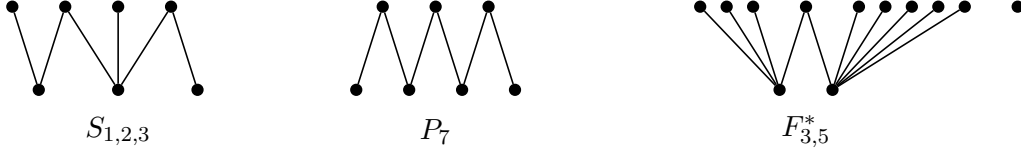


Figure 6: The bipartite graphs from Definition 5.27

Definition 5.27 ($S_{1,2,3}$, P_7 , $F_{p,q}^*$). See Figure 6 for an illustration.

- (1) $S_{1,2,3}$ is the (colored) bipartite graph obtained from a star with three leaves by subdividing one of its edges once and subdividing another edge twice.
- (2) P_7 is the (colored) path on 7 vertices.
- (3) $F_{p,q}^*$ is the colored bipartite graph with vertex color classes $\{a, b\}$ and $\{a_1, \dots, a_p, c, b_1, \dots, b_q, d\}$. The edges are $\{(a, a_i) \mid i \in [p]\}$, $\{(b, b_j) \mid j \in [q]\}$, and $(a, c), (b, c)$.

Combining results due to Allen [All09] (for the $S_{1,2,3}$ and $F_{p,q}^*$ cases) and a result of Lozin & Zamaraev [LZ17] (for the P_7 case), we formally state

Theorem 5.28 ([All09, LZ17]). *Let H be a colored bipartite graph, and let \mathcal{F} be the class of H -free bipartite graphs. If \mathcal{F} has at most factorial speed, then \mathcal{F} is a subclass of either the $S_{1,2,3}$ -free bipartite graphs, the P_7 -free bipartite graphs, or the $F_{p,q}^*$ -free bipartite graphs, for some $p, q \in \mathbb{N}$.*

By the above result, in order to establish Theorem 5.23, it suffices to consider the maximal monogenic factorial classes of bipartite graphs defined by the forbidden induced subgraphs $S_{1,2,3}$, P_7 , $F_{p,q}^*$.

5.2.3 $S_{1,2,3}$ -Free Bipartite Graphs

In this section, we derive Theorem 5.23 for the class of $S_{1,2,3}$ -free bipartite graphs. It is known that the class of $S_{1,2,3}$ -free bipartite graphs has bounded clique-width [LV08], and hence it has also bounded twin-width [BKTW20]. Therefore the following theorem follows immediately from our result for graph classes of bounded twin-width (Theorem 4.1).

Theorem 5.29. *Let \mathcal{F} be a stable class of $S_{1,2,3}$ -free bipartite graphs. Then \mathcal{F} admits a constant-size equality-based adjacency labeling scheme, and hence $\text{SK}(\mathcal{F}_n) = O(1)$.*

5.2.4 $F_{p,q}^*$ -Free Bipartite Graphs

In this section, we prove Theorem 5.23 for classes of $F_{p,q}^*$ -free bipartite graphs by developing a constant-size equality-based adjacency labeling scheme for stable classes of $F_{p,q}^*$ -free bipartite graphs via a sequence of labeling schemes for special subclasses each generalizing the previous one.

We denote by $F_{p,q}$ the bipartite graph with parts $\{a, b\}$ and $\{c, a_1, \dots, a_p, b_1, \dots, b_q\}$, and with edges $(a, c), (b, c), \{(a, a_i) \mid i \in [p]\}, \{(b, b_i) \mid i \in [q]\}$. We also denote by T_p the bipartite graph on

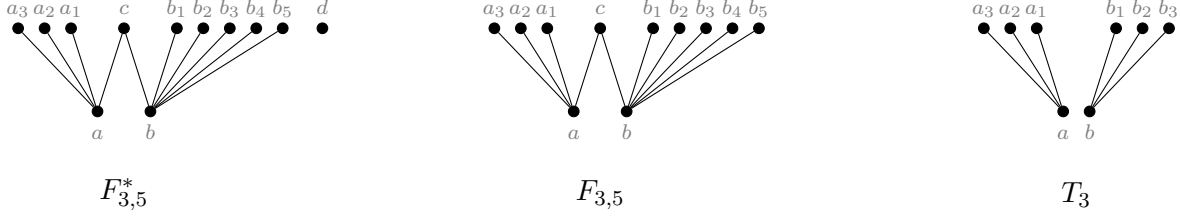


Figure 7: The bipartite graphs considered in Section 5.2.4.

vertex sets $\{a, b\}, \{a_1, \dots, a_p, b_1, \dots, b_p\}$, where (a, a_i) and (b, b_i) are edges for each $i \in [p]$. So T_p is the disjoint union of two stars with $p + 1$ vertices.

Definition 5.30. For $q, s \in \mathbb{N}$ we denote by $Z_{q,s}$ the bipartite graph (X, Y, E) with $|X| = q, |Y| = qs$, where $X = \{x_1, \dots, x_q\}$, Y is partitioned into q sets $Y = Y_1 \cup \dots \cup Y_q$ each of size s , and for every $i \in [q]$:

- (1) x_i is adjacent to all vertices in Y_j for all $1 \leq j \leq i$, and
- (2) x_i is adjacent to no vertices in Y_j for all $i < j \leq q$.

Note that $Z_{q,s}$ is obtained from $H_q^{\circ\circ}$ by duplicating every vertex in one of the parts $s - 1$ times. In particular, $H_q^{\circ\circ}$ is an induced subgraph of $Z_{q,s}$.

We start with structural results and an equality-based labeling scheme for *one-sided* T_p -free bipartite graphs. A colored bipartite graph $G = (X, Y, E)$ is one-sided T_p -free if it does not contain T_p as an induced subgraph such that the centers of both stars belong to X . Note that any T_p -free bipartite graph is also a one-sided T_p -free graph.

Proposition 5.31. *Let $G = (X, Y, E)$ be any one-sided T_p -free bipartite graph and let $u, v \in X$ satisfy $\deg(u) \leq \deg(v)$. Then $|N(u) \cap N(v)| > |N(u)| - p$.*

Proof. For contradiction, assume $|N(u) \cap N(v)| \leq |N(u)| - p$ so that $|N(u) \setminus N(v)| \geq p$. Then since $\deg(v) \geq \deg(u)$ it follows that $|N(v) \setminus N(u)| \geq p$. But then T_p is induced by $\{u, v\}$ and $(N(u) \setminus N(v)) \cup (N(v) \setminus N(u))$. \square

Proposition 5.32. *Suppose $S_1, \dots, S_t \subseteq [n]$ each have $|S_i| \geq n - p$ where $n > pt$. Then $|\bigcap_{j=1}^t S_j| \geq n - pt$.*

Proof. Let R be the set of all $i \in [n]$ such that for some $S_j, i \notin S_j$. Then

$$|R| \leq \sum_{j=1}^t (n - |S_j|) \leq \sum_{j=1}^t p = pt,$$

so $|\bigcap_{j=1}^t S_j| \geq n - |R| \geq n - pt$. \square

Lemma 5.33. *Fix any constants k, q, p such that $k \geq qp + 1$ and let $G = (X, Y, E)$ be any one-sided T_p -free bipartite graph. Then there exists $m \geq 0$ and partitions $X = A_0 \cup A_1 \cup \dots \cup A_m$ and $Y = B_1 \cup \dots \cup B_m \cup B_{m+1}$, where $A_i \neq \emptyset, B_i \neq \emptyset$ for every $i \in [m]$, such that the following hold*

- (1) $|B_i| \geq k$, for all $i \in [m]$.
- (2) For every $j \in \{0, 1, \dots, m\}$, every $x \in A_j$ has less than k neighbours in $\bigcup_{i \geq j+1} B_i$.

- (3) For every $i, j, 1 \leq i \leq j \leq m$, every $x \in A_j$ has more than $|B_i| - p$ neighbours in B_i .
- (4) If $m \geq q$, then $Z_{q, k-pq}$ is an induced subgraph of G .

Proof. Let A_0 be the set of vertices in X that have less than k neighbours. If $A_0 = X$, then $m = 0$, A_0 , and $B_1 = Y$ satisfy the conditions of the lemma. Otherwise, we construct the remaining parts of partitions using the following procedure. Initialize $X' = X \setminus A_0$, $Y' = Y$, and $i = 1$.

1. Let a_i be a vertex in X' with the least number of neighbours in Y' .
2. Let B_i be the set of all neighbors of a_i in $G[X', Y']$.
3. Let A_i be the set of vertices in X' with degree less than k in $G[X', Y' \setminus B_i]$. Note that A_i contains a_i .
4. $X' \leftarrow X' \setminus A_i$, $Y' \leftarrow Y' \setminus B_i$.
5. If $X' = \emptyset$, then $B_{i+1} = Y'$, let $m = i$, and terminate the procedure; Otherwise increment i and return to step 1.

Conditions (1) and (2) follow by definition. Next we will prove condition (3) by showing that for every $1 \leq i \leq j \leq m$, every $x \in A_j$ has more than $|B_i| - p$ neighbours in B_i . Suppose, towards a contradiction, that $|N(x) \cap B_i| \leq |B_i| - p$. Consider X', Y' as in round i of the construction procedure, so B_i is the neighbourhood of a_i in $G[X', Y']$. Then x has degree at least that of a_i in $G[X', Y']$, and hence the conclusion holds by Proposition 5.31.

Finally, to prove condition (4) we will show that for any $q \leq m$ there exist sets $B'_1 \subseteq B_1, \dots, B'_q \subseteq B_q$ so that the vertices $\{a_1, \dots, a_q\}$ and the sets B'_1, \dots, B'_q induce $Z_{q, k-pq}$. First, observe that by construction for every $1 \leq i < j \leq m$, a_i has no neighbours in B_j . Now, let $i \in [m]$, then by condition (3), for all $i \leq j \leq m$ it holds that $|N(a_j) \cap B_i| > |B_i| - p$. Since $|B_i| \geq k > pq$, it holds by Proposition 5.32 that

$$\left| B_i \cap \bigcap_{j=i}^q N(a_j) \right| \geq |B_i| - pq \geq k - pq.$$

We define $B'_i = B_i \cap \bigcap_{j=i}^q N(a_j)$. Then for each $i \in [m]$ it holds that a_i is adjacent to all vertices in B'_j for all $1 \leq j \leq i$, but a_i is adjacent to no vertices in B'_j for $i < j \leq m$. Hence the vertices $\{a_1, \dots, a_q\}$ and the sets B'_1, \dots, B'_q induce $Z_{q, k-pq}$, which proves condition (4) and concludes the proof of the lemma. \square

Lemma 5.34. *Let $p \in \mathbb{N}$ and let \mathcal{T} be a stable class of one-sided T_p -free bipartite graphs. Then \mathcal{T} admits a constant-size equality-based adjacency labeling scheme, and hence $\text{SK}(\mathcal{T}_n) = O(1)$.*

Proof. Since \mathcal{T} is stable, it does not contain $\mathcal{C}^{\circ\circ}$ as a subclass. Let q be the minimum number such that $H_q^{\circ\circ} \notin \mathcal{T}$, and let $G = (X, Y, E)$ be an arbitrary graph from \mathcal{T} .

Let $k = qp + 1$ and let $X = A_0 \cup A_1 \cup \dots \cup A_m$ and $Y = B_1 \cup \dots \cup B_m \cup B_{m+1}$ be partitions satisfying the conditions of Lemma 5.33. Since G does not contain $H_q^{\circ\circ}$ as an induced subgraph, it holds that $m < q$.

We construct the labels for the vertices of G as follows. For a vertex $x \in X$ we define $\ell(x)$ as a label consisting of several tuples. The first tuple is $(0, i \mid -)$, where $i \in \{0, 1, \dots, m\}$ is the unique index such that $x \in A_i$. This tuple follows by i tuples $(- \mid y_1^j, y_2^j, \dots, y_{p_j}^j)$, $j \in [i]$, where $p_j < p$ and $\{y_1^j, y_2^j, \dots, y_{p_j}^j\}$ are the non-neighbours of x in B_j . The last tuple of $\ell(x)$ is $(- \mid y_1^{i+1}, y_2^{i+1}, \dots, y_k^{i+1})$,

where $k' < k$ and $y_1^{i+1}, y_2^{i+1}, \dots, y_{k'}^{i+1}$ are the neighbours of x in $\bigcup_{i \geq j+1} B_i$. For a vertex $y \in Y$ we define $\ell(y) = (1, i \mid y)$, where $i \in [m+1]$ is the unique index such that $y \in B_i$.

Note that, in every label, the total length of prefixes is at most $1 + \lceil \log m \rceil \leq 1 + \lceil \log q \rceil$, and the total number of equality codes depends only on p, q , and k , which are constants. Therefore it remains to show that the labels can be used to define an equality-based adjacency decoder.

Given two vertices x, y in G the decoder operates as follows. First, it checks the first prefixes in the first tuples of $\ell(x)$ and $\ell(y)$. If they are the same, then x, y belong to the same part in G and the decoder outputs 0. Hence, we can assume that they are different. Without loss of generality, let $\ell(x) = (0, i \mid -)$ and $\ell(y) = (1, j \mid y)$, so $x \in A_i \subseteq X$ and $y \in B_j \subseteq Y$.

If $j \leq i$, then the decoder compares y with the equality codes $y_1^j, y_2^j, \dots, y_{p_j}^j$ of the $(j+1)$ -th tuple of $\ell(x)$. If y is equal to at least one of them, then y is among the non-neighbours of x in B_j and the decoder outputs 0; otherwise, x and y are adjacent and the decoder outputs 1. If $j > i$, then the decoder compares y with the equality codes $y_1^{i+1}, y_2^{i+1}, \dots, y_{k'}^{i+1}$ of the last tuples of $\ell(x)$, and if y is equal to at least one of them, then y is among the neighbours of x in $\bigcup_{i \geq j+1} B_i$ and the decoder outputs 1; otherwise, x and y are not adjacent and the decoder outputs 0. \square

Next, we develop an equality-based labeling scheme for stable classes of one-sided $F_{p,p}$ -free bipartite graphs. A colored bipartite graph $G = (X, Y, E)$ is one-sided $F_{p,p}$ -free if it does not contain $F_{p,p}$ as an induced subgraph such that the part of $F_{p,p}$ of size 2 is a subset of X .

Proposition 5.35. *Let $G = (X, Y, E)$ be any one-sided $F_{p,p}$ -free bipartite graph and let $u, v \in X$ satisfy $\deg(u) \leq \deg(v)$. Then either $N(u) \cap N(v) = \emptyset$ or $|N(u) \cap N(v)| > |N(u)| - p$.*

Proof. Suppose that $N(u) \cap N(v) \neq \emptyset$, and for contradiction assume that $|N(u) \setminus N(v)| \geq p$. Since $\deg(u) \leq \deg(v)$, this means $|N(v) \setminus N(u)| \geq |N(u) \setminus N(v)| \geq p$. Let $w \in N(u) \cap N(v)$. Then $\{u, v\}$ with $\{w\} \cup (N(v) \setminus N(u)) \cup (N(u) \setminus N(v))$ induces a graph containing $F_{p,p}$, a contradiction. \square

Proposition 5.36. *Let $G = (X, Y, E)$ be any one-sided $F_{p,p}$ -free bipartite graph and let $x, y, z \in X$ satisfy $\deg(x) \geq \deg(y) \geq \deg(z) \geq 2p$. Suppose that $N(y) \cap N(z) \neq \emptyset$. Then*

$$N(x) \cap N(y) = \emptyset \iff N(x) \cap N(z) = \emptyset.$$

Proof. Since $N(y) \cap N(z) \neq \emptyset$, it holds that $|N(y) \cap N(z)| > |N(z)| - p \geq p$ by Proposition 5.35.

Suppose that $N(x) \cap N(y) \neq \emptyset$. For contradiction, assume that $N(x) \cap N(y) \cap N(z) = \emptyset$. Then $|N(y) \setminus N(x)| \geq |N(y) \cap N(z)| > |N(z)| - p \geq p$, which contradicts $|N(y) \cap N(x)| > |N(y)| - p$.

Now suppose that $N(x) \cap N(y) = \emptyset$. For contradiction, assume that $N(x) \cap N(z) \neq \emptyset$. Then $|N(x) \cap N(z)| \leq |N(z) \setminus N(y)| < p \leq |N(z)| - p < |N(x) \cap N(z)|$, a contradiction. \square

We will say that a bipartite graph $G = (X, Y, E)$ is *left-disconnected* if there are two vertices $x, y \in X$ that are in different connected components of G . It is *left-connected* otherwise.

Proposition 5.37. *Let $G = (X, Y, E)$ be any one-sided $F_{p,p}$ -free bipartite graph where every vertex in X has degree at least $2p$. Let $x \in X$ have maximum degree of all vertices in X . If G is left-connected, then for any $y \in X$ it holds that $|N(y) \cap N(x)| > |N(y)| - p$.*

Proof. Let $y \in X$. Since G is left-connected, there is a path from y to x . Let y_0, y_1, \dots, y_t be the path vertices in X , where $y = y_0, x = y_t$, and $N(y_{i-1}) \cap N(y_i) \neq \emptyset$ for each $i \in [t]$. By Propositions 5.36 and 5.35, it holds that if $N(y_i) \cap N(x) \neq \emptyset$ then $|N(y_i) \cap N(x)| > |N(y_i)| - p$ and $|N(y_{i-1}) \cap N(x)| > |N(y_{i-1})| - p$. Therefore the conclusion holds, because $N(y_{t-1}) \cap N(x) = N(y_{t-1}) \cap N(y_t) \neq \emptyset$. \square

Lemma 5.38. *Fix any constants $p, q \geq 1$, let $k = (q+1)p$, and let $G = (X, Y, E)$ be any connected one-sided $F_{p,p}$ -free bipartite graph. Then there exists a partition $X = X_0 \cup X_1 \cup X_2$ (where some of the sets can be empty) such that the following hold:*

- (1) X_0 is the set of vertices in X that have degree less than k .
- (2) The induced subgraph $G[X_1, Y]$ is one-sided T_p -free.
- (3) The induced subgraph $G[X_2, Y]$ is left-disconnected.
- (4) For any r, s such that $r < q$ and $p < s \leq k$, if $X_1 \neq \emptyset$ and $Z_{r,s} \sqsubset G[X_2, Y]$, then $Z_{r+1, s-p} \sqsubset G$.

Proof. Let X_0 be the set of vertices in X that have degree less than k , and let $X' = X \setminus X_0$. If $G[X', Y]$ is left-disconnected, then we define $X_1 = \emptyset$ and $X_2 = X'$.

Assume now that $G[X', Y]$ is left-connected. By Proposition 5.37, the highest-degree vertex $x \in X'$ satisfies $|N(x) \cap N(y)| > |N(y)| - p$ for every $y \in X'$. Define X_1 as follows: add the highest-degree vertex x to X_1 , and repeat until $G[X' \setminus X_1, Y]$ is left-disconnected. Then set $X_2 = X' \setminus X_1$. Condition 3 holds by definition, so it remains to prove conditions 2 and 4.

For every $a, b \in X_1$, note that $N(a) \cap N(b) \neq \emptyset$. Suppose for contradiction that $T_p \sqsubset G[X_1, Y]$, then there are $a, b \in X_1$ such that T_p is contained in the subgraph induced by the vertices $\{a, b\}$ and $(N(a) \setminus N(b)) \cup (N(b) \setminus N(a))$. But then adding any $c \in N(a) \cap N(b)$ results in a forbidden copy of induced $F_{p,p}$, a contradiction. This proves condition 2.

Now for any r, s such that $r < q$ and $p < s \leq k$, suppose that $X_1 \neq \emptyset$ and $Z_{r,s} \sqsubset G[X_2, Y]$. Then there are $u_1, \dots, u_r \in X_2$ and pairwise disjoint sets $V_1 \subseteq N(u_1), \dots, V_r \subseteq N(u_r)$ such that for each i , $|V_i| = s$, for every $1 \leq j \leq i$, v_i is adjacent to all vertices in V_j , and for every $i < j \leq r$, v_i is adjacent to no vertices in V_j .

Let x be the vertex in X_1 with least degree, so that x was the last vertex to be added to X_1 . Then $G[X_2 \cup \{x\}, Y]$ is left-connected but $G[X_2, Y]$ is left-disconnected, and x is the highest-degree vertex of $G[X_2 \cup \{x\}, Y]$ in $X_2 \cup \{x\}$. Since u_1, \dots, u_r are in the same connected component of $G[X_2, Y]$, but the graph $G[X_2, Y]$ is disconnected, it must be that there is $z \in X_2$ such that $N(z) \cap N(u_i) = \emptyset$ for all u_i . It is also the case that $|N(x) \cap N(z)| > |N(z)| - p \geq k - p \geq s - p$ by Proposition 5.37, since x has the highest degree in $X_2 \cup \{x\}$.

Observe that for each $V_i \subseteq N(u_i)$ it holds that $|N(x) \cap V_i| \geq s - p$ also by Proposition 5.37. Set $V'_i = V_i \cap N(x)$ for each $i \in [r]$, and set $V'_{r+1} = N(x) \cap N(z)$. Clearly, the graph induced by $\{u_1, \dots, u_r, x\} \cup V'_1 \cup V'_2 \cup \dots \cup V'_r \cup V'_{r+1}$ contains $Z_{r+1, s-p}$ as an induced subgraph. \square

We will now use the above structural result to construct a suitable decomposition scheme for stable one-sided $F_{p,p}$ -free bipartite graphs. Let $p, q \geq 1$ be fixed constants, let $k = (q+1)p$, and let $\mathcal{F}_{p,q}$ be the class of one-sided $F_{p,p}$ -free bipartite graphs that do not contain $H_q^{\circ\circ}$ as an induced subgraph. Let $G = (X, Y, E) \in \mathcal{F}_{p,q}$. Using Lemma 5.38, we define a decomposition tree \mathcal{T} for G inductively as follows. Let G_v be the induced subgraph of G associated with node v of the decomposition tree and write $X' \subseteq X$, $Y' \subseteq Y$ for its sets of vertices, so $G_v = G[X', Y']$. Graph G is associated with the root node of \mathcal{T} .

- If G_v is one-sided T_k -free, terminate the decomposition, so v is a leaf node (L -node) of the decomposition tree.
- If G_v is disconnected (in particular, if it is left-disconnected), then v is a D -node such that the children are the connected components of G_v .

- If G_v is connected and not one-sided T_k -free, then X' admits a partition $X' = X'_0 \cup X'_1 \cup X'_2$ satisfying the condition of [Lemma 5.38](#). Since G_v is connected, $X'_0 \cup X'_1 \neq \emptyset$. Furthermore, since G_v is not one-sided T_k -free, $X'_2 \neq \emptyset$. Hence, v is a P -node with exactly two children v_1 and v_2 , where $G_{v_1} = G[X'_0 \cup X'_1, Y']$ and $G_{v_2} = G[X'_2, Y']$. Observe that

- (1) G_{v_1} is one-sided T_k -free, and therefore v_1 is a leaf;
- (2) G_{v_2} is left-disconnected, and therefore v_2 is a D -node; furthermore, every vertex $x \in X'_2$ has degree at least k in G_{v_2} (otherwise it would be included in the set X'_0).

Proposition 5.39. *Let \mathcal{Q} be the class of one-sided T_k -free bipartite graphs. Then the graphs in $\mathcal{F}_{p,q}$ admit $(\mathcal{Q}, 2)$ -decomposition trees of depth at most $2q$.*

Proof. By definition, the above decomposition scheme produces $(\mathcal{Q}, 2)$ -decomposition trees. In the rest of the proof we will establish the claimed bound on the depth of any such tree. Suppose, towards a contradiction, that there exists a graph $G = (X, Y, E) \in \mathcal{F}_{p,q}$ such that the decomposition tree \mathcal{T} for G has depth at least $2q + 1$. Let $\mathcal{P} = (v_0, v_1, v_2, \dots, v_s)$ be a leaf-to-root path in \mathcal{T} of length $s \geq 2q + 1$, where v_0 is a leaf and v_s is the root. Denote by $G_{v_i} = G[X^i, Y^i]$ the graph corresponding to a node v_i in \mathcal{P} . By construction, all internal nodes of \mathcal{P} are either D -nodes or P -nodes. Clearly, the path cannot contain two consecutive D -nodes, as any child of a D -node is a connected graph. Furthermore, a unique non-leaf child v_i of a P -node is a D -node, and every $x \in X^i$ has degree at least k in G_{v_i} . Consequently, P -nodes and D -nodes alternated along (the internal part of) \mathcal{P} .

Let $v_{i-1}, v_i, v_{i+1}, v_{i+2}$ be four internal nodes of \mathcal{P} , where v_{i-1} and v_{i+1} are D -nodes, and v_i and v_{i+2} are P -nodes. Recall that, since the parent v_{i+2} of v_{i+1} is a P -node, every vertex in X_{i+1} has degree at least k in $G_{v_{i+1}}$. Hence, since G_{v_i} is a connected component of $G_{v_{i+1}}$, every vertex in $X^i \subseteq X^{i+1}$ also has degree at least k . Let $X^i = X_0^i \cup X_1^i \cup X_2^i$ be the partition of X^i according to the decomposition rules. Since $X_0^i \cup X_1^i \neq \emptyset$ and $X_0^i = \emptyset$, we conclude that $X_1^i \neq \emptyset$. Therefore, by [Lemma 5.38](#), if the $G_{v_{i-1}} = G[X_2^i, Y^i]$ contains $Z_{r,s}$ for some $r < q$ and $p < s \leq k$, then G_{v_i} contains $Z_{r+1, s-p}$.

Let v_t be the first D -node in \mathcal{P} . Note that $t \leq 2$. Every vertex in X^t has degree at least k in G_{v_t} , and therefore $Z_{1,k} \sqsubset G_{v_t}$. By induction, the above discussion implies that for $1 \leq i \leq q - 1$, the graph $Z_{1+i, k-ip}$ is an induced subgraph of $G_{v_{t+2i-1}}$. Hence, since the length of \mathcal{P} is at least $2q + 1$, we have $H_q^{\circ\circ} = Z_{q,1} \sqsubset Z_{q, k-(q-1)p} \sqsubset G_{v_{t+2q-3}} \sqsubset G$, a contradiction. \square

Lemma 5.40. *Let $p \in \mathbb{N}$ and let \mathcal{F} be a stable class of one-sided $F_{p,p}$ -free bipartite graphs. Then \mathcal{F} admits a constant-size equality-based adjacency labeling scheme, and hence $\text{SK}(\mathcal{F}_n) = O(1)$.*

Proof. Since \mathcal{F} is stable, it does not contain $\mathcal{C}^{\circ\circ}$ as a subclass. Let q be the minimum number such that $H_q^{\circ\circ} \notin \mathcal{F}$. Let $k = (q+1)p$ and let \mathcal{Q} be the class of one-sided T_k -free bipartite graphs. We have that $\mathcal{F} \subseteq \mathcal{F}_{p,q}$, and therefore, by [Proposition 5.39](#), the graphs in \mathcal{F} admit $(\mathcal{Q}, 2)$ -decomposition trees of depth at most $2q$. Hence, by [Lemma 5.34](#) and [Lemma 5.25](#), \mathcal{F} admits a constant-size equality-based adjacency labeling scheme. \square

We conclude this section by showing that stable classes of $F_{p,p}^*$ -free graphs admit constant-size equality-based adjacency labeling schemes. For this we will use the above result for one-sided $F_{p,p}$ -free graphs and the following

Proposition 5.41 ([\[All09\]](#), Corollary 9). *Let $G = (X, Y, E)$ be a $F_{p,p}^*$ -free bipartite graph. Then there is a partition $X = X_1 \cup X_2$ and $Y = Y_1 \cup Y_2$, where $|Y_2| \leq 1$, such that both $G[X_1, Y_1]$ and $G[X_2, Y_1]$ are one-sided $F_{p,p}$ -free.*

Theorem 5.42. *For any constants $p, p' \geq 1$, a stable class \mathcal{F} of $F_{p,p'}^*$ -free bipartite graphs admits a constant-size equality-based adjacency labeling scheme, and hence $\text{SK}(\mathcal{F}_n) = O(1)$.*

Proof. As before, since \mathcal{F} is stable, it does not contain $\mathcal{C}^{\circ\circ}$ as a subclass. Let q be the minimum number such that $H_q^{\circ\circ} \notin \mathcal{F}$, and assume without loss of generality that $p \geq p'$. It follows that \mathcal{F} is a subclass of $(F_{p,p}^*, H_q^{\circ\circ})$ -free bipartite graphs. Let $G = (X, Y, E)$ be a member of this class. Let $X = X_1 \cup X_2, Y = Y_1 \cup Y_2$ be the partition given by Proposition 5.41. We assign labels as follows.

We start the label for each vertex with a one-bit prefix indicating whether it is in X or Y . We then append the following labels. For $x \in X$, we use another one-bit prefix that is equal to 1 if x is adjacent to the unique vertex Y_2 , and 0 otherwise. Then, we use one more one-bit prefix to indicate whether $x \in X_1$ or $x \in X_2$. If $x \in X_1$, complete the label by using the labeling scheme of Lemma 5.40 for $G[X_1, Y_1]$. If $x \in X_2$, complete the label by using the labeling scheme of Lemma 5.40 for $\overline{G[X_2, Y_1]}$.

For $y \in Y$, use a one-bit prefix to indicate whether $y \in Y_2$. If $y \in Y_1$ then concatenate the two labels for y obtained from the labeling scheme of Lemma 5.40 for $G[X_1, Y_1]$ and $\overline{G[X_2, Y_1]}$.

The decoder first checks if x, y are in opposite parts. Now assume $x \in X, y \in Y$. The decoder checks if $y \in Y_2$ and outputs the appropriate value using the appropriate prefix from the label of x . Then if $x \in X_1$, it uses the labels of x and y in $G[X_1, Y_1]$; otherwise it uses the labels of x and y in $\overline{G[X_2, Y_1]}$ and flips the output. \square

5.2.5 P_7 -Free Bipartite Graphs

In this section, we prove Theorem 5.23 for P_7 -free bipartite graphs by developing a constant-size equality-based adjacency labeling scheme for stable classes of P_7 -free bipartite graphs

In the below definition, for two disjoint sets of vertices A and B we say that A is *complete* to B if every vertex in A is adjacent to every vertex in B ; we also say that A is *anticomplete* to B if there are no edges between A and B .

Definition 5.43 (Chain Decomposition). See Figure 8 for an illustration of the chain decomposition. Let $G = (X, Y, E)$ be a bipartite graph and $k \in \mathbb{N}$. We say that G admits a *k -chain decomposition* if one of the parts, say X , can be partitioned into subsets $A_1, \dots, A_k, C_1, \dots, C_k$ and the other part Y can be partitioned into subsets $B_1, \dots, B_k, D_1, \dots, D_k$ in such a way that:

- For every $i \leq k - 1$, the sets A_i, B_i, C_i, D_i are non-empty. For $i = k$, at least one of the sets A_i, B_i, C_i, D_i must be non-empty.
- For each $i = 1, \dots, k$,
 - every vertex of B_i has a neighbour in A_i ;
 - every vertex of D_i has a neighbour in C_i ;
- For each $i = 2, \dots, k - 1$,
 - every vertex of A_i has a non-neighbour in B_{i-1} ;
 - every vertex of C_i has a non-neighbour in D_{i-1} ;
- For each $i = 1, \dots, k$,
 - the set A_i is anticomplete to B_j for $j > i$ and is complete to B_j for $j < i - 1$;
 - the set C_i is anticomplete to D_j for $j > i$ and is complete to D_j for $j < i - 1$;

- For each $i = 1, \dots, k$,
 - the set A_i is complete to D_j for $j < i$, and is anticomplete to D_j for $j \geq i$;
 - the set C_i is complete to B_j for $j < i$, and is anticomplete to B_j for $j \geq i$.

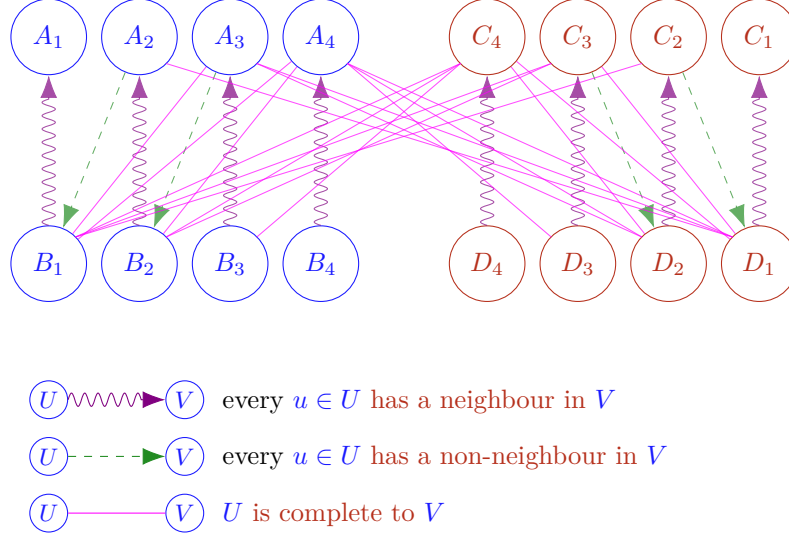


Figure 8: Example of a 4-chain decomposition.

Remark 5.44. In the case of a 2-chain decomposition of a connected P_7 -free bipartite graphs, we will also need the fact that every vertex in A_2 and every vertex in A_1 have a neighbour in common; and every vertex in C_2 and every vertex in C_1 have a neighbour in common. This is not stated explicitly in [LZ17], but easily follows from a proof in [LZ17]. Since the neighbourhood of every vertex in A_1 lies entirely in B_1 , the above fact implies that every vertex in A_2 has a neighbour in B_1 . Similarly, the neighbourhood of every vertex in C_1 lies entirely in D_1 , and therefore every vertex in C_2 has a neighbour in D_1 .

Theorem 5.45 ([LZ17]). *Let $G = (X, Y, E)$ be a P_7 -free bipartite graph such that both G and \overline{G} are connected. Then G or \overline{G} admits a k -chain decomposition for some $k \geq 2$.*

Lemma 5.46. *Let $G = G(X, Y, E)$ be a connected P_7 -free bipartite graph of chain number c that admits a k -chain decomposition for some $k \geq 2$. Then there exists a partition of X into $p \leq 2(c+1)$ sets X_1, X_2, \dots, X_p , and a partition of Y into $q \leq 2(c+1)$ sets Y_1, Y_2, \dots, Y_q such that, for any $i \in [p], j \in [q]$,*

$$\text{ch}(G[X_i, Y_j]) < \text{ch}(G).$$

Proof. Assume, without loss of generality, that X is partitioned into subsets $A_1, \dots, A_k, C_1, \dots, C_k$ and Y is partitioned into subsets $B_1, \dots, B_k, D_1, \dots, D_k$ satisfying Definition 5.43. Since at least one of the sets A_k, B_k, C_k, D_k is non-empty, and every vertex in B_k has a neighbour in A_k and every vertex in D_k has a neighbour in C_k , at least one of A_k and C_k is non-empty. Without loss of generality we assume that A_k is not empty. It is straightforward to check by definition that for any vertices $a_2 \in A_2, a_3 \in A_3, \dots, a_k \in A_k$, and $d_1 \in D_1, d_2 \in D_2, \dots, d_{k-1} \in D_{k-1}$ the subgraph of G induced by $\{a_2, a_3, \dots, a_k, d_1, d_2, \dots, d_{k-1}\}$ is isomorphic to $H_{k-1}^{\circ\circ}$, which implies that k is at most $c+1$. We also observe that any path from a vertex in A_1 to a vertex in D_1 contains at least 4 vertices, and hence G contains $H_2^{\circ\circ}$ and $\text{ch}(G) \geq 2$. We split our analysis in two cases.

Case 1. $k \geq 3$. We will show that for any $X' \in \{A_1, \dots, A_k, C_1, \dots, C_k\}$ and $Y' \in \{B_1, \dots, B_k, D_1, \dots, D_k\}$, $\text{ch}(G[X', Y']) < \text{ch}(G)$. Since $\text{ch}(G) \geq 2$, the chain number of a biclique is 1, and the chain number of a co-biclique is 0, we need only to verify pairs of sets that can induce a graph which is neither a biclique nor a co-biclique. By [Definition 5.43](#), these are the pairs (A_i, B_i) , (C_i, D_i) for $i \in [k]$ and (A_i, B_{i-1}) , (C_i, D_{i-1}) for $i \in \{2, \dots, k\}$.

We start with the pair (A_1, B_1) . Since D_2 is anticomplete to A_1 , and C_2 is complete to B_1 , for any vertex $d_2 \in D_2$ and its neighbour $c_2 \in C_2$ we have that $\text{ch}(G[A_1, B_1]) < \text{ch}(G[A_1 \cup \{c_2\}, B_1 \cup \{d_2\}]) \leq \text{ch}(G)$. Similarly, since D_1 is complete to all A_2, A_3, \dots, A_k , and C_1 is anticomplete to all B_1, B_2, \dots, B_k , addition of a vertex $d_1 \in D_1$ and its neighbour $c_1 \in C_1$ to any of the graphs $G[A_i, B_i]$ or $G[A_i, B_{i-1}]$ for $i \in \{2, \dots, k\}$ strictly increases the chain number of that graph. Symmetric arguments establish the desired conclusion for the pairs of sets (C_i, D_i) , $i \in [k]$, and (C_i, D_{i-1}) , $i \in \{2, \dots, k\}$.

In this case, $A_1, \dots, A_k, C_1, \dots, C_k$ and $B_1, \dots, B_k, D_1, \dots, D_k$ are the desired partitions of X and Y respectively.

Case 2. $k = 2$. Assume first that both A_2 and C_2 are non-empty. Let c_2 be a vertex in C_2 , d_1 be a neighbour of c_2 in D_1 (which exists by [Remark 5.44](#)), and c_1 be a neighbour of d_1 in C_1 . Since C_2 is complete to B_1 and D_1 is anticomplete to A_1 , $\text{ch}(G[A_1, B_1]) < \text{ch}(G[A_1 \cup \{c_2\}, B_1 \cup \{d_1\}]) \leq \text{ch}(G)$. Similarly, because C_1 is anticomplete to both B_1 and B_2 and D_1 is complete to A_2 , we have that $\text{ch}(G[A_2, B_1]) < \text{ch}(G[A_2 \cup \{c_1\}, B_1 \cup \{d_1\}]) \leq \text{ch}(G)$ and $\text{ch}(G[A_2, B_2]) < \text{ch}(G[A_2 \cup \{c_1\}, B_2 \cup \{d_1\}]) \leq \text{ch}(G)$. Using symmetric arguments we can show that the chain number of each of $G[C_1, D_1]$, $G[C_2, D_1]$, and $G[C_2, D_2]$ is strictly less than the chain number of G . All other pairs of sets (X', Y') , where $X' \in \{A_1, A_2, C_1, C_2\}$ and $Y' \in \{B_1, B_2, D_1, D_2\}$ induce either a biclique or a co-biclique, and therefore $\text{ch}(G[X', Y']) < \text{ch}(G)$. In this case, A_1, A_2, C_1, C_2 and B_1, B_2, D_1, D_2 are the desired partitions of X and Y respectively.

The case when one of A_2 and C_2 is empty requires a separate analysis. Assume that $A_2 \neq \emptyset$ and $C_2 = \emptyset$. The case when $A_2 = \emptyset$ and $C_2 \neq \emptyset$ is symmetric and we omit the details. Since C_2 is empty, D_2 is also empty and therefore A_1, A_2, C_1 is a partition of X , and B_1, B_2, D_1 is a partition of Y . Let a_2 be a vertex in A_2 , d_1 be a vertex in D_1 , and c_1 be a neighbour of d_1 in C_1 . Let B'_1 be the neighbourhood of a_2 in B_1 and let $B''_1 = B_1 \setminus B'_1$. We claim that A_1, A_2, C_1 and B'_1, B''_1, B_2, D_1 are the desired partitions of X and Y respectively. All the pairs of sets, except (A_1, B'_1) and (A_1, B''_1) , can be treated as before and we skip the details. For (A_1, B'_1) , we observe that a_2 is complete to B'_1 and D_1 is anticomplete to A_1 , and hence $\text{ch}(G[A_1, B'_1]) < \text{ch}(G[A_1 \cup \{a_2\}, B'_1 \cup \{d_1\}]) \leq \text{ch}(G)$.

To establish the desired property for (A_1, B''_1) , we first observe that by [Remark 5.44](#) every vertex in A_1 has a neighbour in common with a_2 , and therefore every vertex in A_1 has a neighbour in B'_1 . If $G[A_1, B''_1]$ is edgeless the property holds trivially. Otherwise, let $P \subseteq A_1$ and $Q \subseteq B''_1$ be such that $P \cup Q$ induces a $H_s^{\circ\circ}$ in $G[A_1, B''_1]$, where $s \geq 1$ is the chain number of the latter graph. Let x be the vertex in P that has degree 1 in $G[P, Q]$, and let y be a neighbour of x in B'_1 . We claim that y is complete to P . Indeed, if y is not adjacent to some $x' \in P$, then $x', z, x, y, a_2, d_1, c_1$ would induce a forbidden P_7 , where z is the vertex in Q that is adjacent to every vertex in P . Consequently, $G[P \cup \{a_2\}, Q \cup \{y\}]$ is isomorphic to $H_{s+1}^{\circ\circ}$, and therefore $\text{ch}(G[A_1, B''_1]) < \text{ch}(G)$. \square

For two pairs of numbers (a, b) and (c, d) we write $(a, b) \preceq (c, d)$ if $a \leq c$ and $b \leq d$, and we write $(a, b) \prec (c, d)$ if at least one of the inequalities is strict.

Lemma 5.47. *Let $G = G(X, Y, E)$ be a P_7 -free bipartite graph such that both G and \bar{G} are connected, and let c be the chain number of G . Then there exists a partition of X into $p \leq 2(c+2)$ sets X_1, X_2, \dots, X_p , and a partition of Y into $q \leq 2(c+2)$ sets Y_1, Y_2, \dots, Y_q such that for any $i \in [p]$,*

$j \in [q]$

$$\left(\text{ch}(G_{i,j}), \text{ch}(\overline{G_{i,j}}) \right) \prec \left(\text{ch}(G), \text{ch}(\overline{G}) \right),$$

where $G_{i,j} = G[X_i, Y_j]$.

Proof. It is easy to verify that for any $k \geq 2$, the graph $\overline{H_k^{\circ\circ}}$ contains the half graph $H_{k-1}^{\circ\circ}$, which implies that $\text{ch}(\overline{G}) \leq \text{ch}(G) + 1 = c + 1$. Furthermore, the bipartite complement of a P_7 is again P_7 , and hence the bipartite complement of any P_7 -free bipartite graph is also P_7 -free.

By [Theorem 5.45](#), G or \overline{G} admits a k -chain decomposition for some $k \geq 2$. Therefore, by [Lemma 5.46](#) applied to either G or \overline{G} , there exist a partition of X into at most $p \leq 2(c+2)$ sets X_1, X_2, \dots, X_p , and a partition of Y into at most $q \leq 2(c+2)$ sets Y_1, Y_2, \dots, Y_q such that either $\text{ch}(G_{i,j}) < \text{ch}(G)$ holds for any $i \in [p], j \in [q]$, or $\text{ch}(\overline{G_{i,j}}) < \text{ch}(\overline{G})$ holds for any $i \in [p], j \in [q]$. This together with the fact that the chain number of an induced subgraph of a graph is never larger than the chain number of the graph, implies the lemma. \square

We are now ready to specify a decomposition scheme for P_7 -free bipartite graphs. Let $G = (X, Y, E)$ be a P_7 -free bipartite graph of chain number c . Let \mathcal{Q} be the class consisting of bicliques and co-bicliques. We define a $(\mathcal{Q}, 2(c+2))$ -decomposition tree \mathcal{T} for G inductively as follows. Let G_v be the induced subgraph of G associated with node v of the decomposition tree and write $X' \subseteq X, Y' \subseteq Y$ for its sets of vertices, so $G_v = G[X', Y']$. Graph G is associated with the root node of \mathcal{T} .

- If G_v belongs to \mathcal{Q} , then terminate the decomposition, so v is a leaf node (L -node) of the decomposition tree.
- If G_v does not belong to \mathcal{Q} and is disconnected, then v is a D -node such that the children are the connected components of G_v .
- If G_v does not belong to \mathcal{Q} , is connected, and $\overline{G_v}$ is disconnected, then v is a \overline{D} -node. There are sets $X'_1, \dots, X'_t \subseteq X'$ and $Y'_1, \dots, Y'_t \subseteq Y'$ such that $\overline{G[X'_1, Y'_1]}, \dots, \overline{G[X'_t, Y'_t]}$ are the connected components of $\overline{G_v}$. The children of this node are $G[X'_1, Y'_1], \dots, G[X'_t, Y'_t]$.
- If G_v does not belong to \mathcal{Q} , and neither G_v , nor $\overline{G_v}$ is disconnected, then v is a P -node. Let X'_1, X_2, \dots, X'_p be a partition of X' into $p \leq 2(c+2)$ sets, and Y'_1, Y_2, \dots, Y'_q be a partition of Y' into $q \leq 2(c+2)$ sets, as in [Lemma 5.47](#). The children of this node are $G[X'_i, Y'_j], i \in [p], j \in [q]$.

Claim 5.48. *Let \mathcal{T} be a decomposition tree as above, and let $G_i = G[X_i, Y_i], i = 1, 2, 3$, be internal nodes in \mathcal{T} such that G_3 is the parent of G_2 which is in turn the parent of G_1 . Then*

- (1) *one of G_3, G_2 , and G_1 is a P -node, or G_i is \overline{D} -node and G_{i-1} is a D -node for some $i \in \{3, 2\}$;*
- (2) *if G_3 is a \overline{D} -node and G_2 is a D -node, then $\text{ch}(G_1) < \text{ch}(G_3)$.*

Proof. We start by proving the first statement. Observe that every child a D -node is a connected graph, and therefore it is not a D -node. Similarly, the bipartite complement of every child of a \overline{D} -node is a connected graph, and therefore a \overline{D} -node cannot have a \overline{D} -node as a child. Hence, if none of G_3, G_2, G_1 is a P -node, either G_3 is a \overline{D} -node and therefore G_2 is a D -node, or G_3 is a D -node, in which case G_2 is a \overline{D} -node and G_1 is a D -node. In both cases we have a pair of parent-child nodes, where the parent is a \overline{D} -node and the child is a D -node.

To prove the second statement, let now G_3 be a \overline{D} -node and G_2 be a D -node, i.e., $G[X_2, Y_2]$ is disconnected, while $G[X_3, Y_3]$ is connected, but its bipartite complement is disconnected. Then

there are sets $X'_1 \subseteq X_2 \setminus X_1$ and $Y'_1 \subseteq Y_2 \setminus Y_1$ such that $G[X'_1, Y'_1]$ and $G[X_1, Y_1]$ are connected components of $G[X_2, Y_2]$ and at least one of X'_1, Y'_1 is non-empty. Also at least one of the sets $X'_2 = X_3 \setminus X_2$ and $Y'_2 = Y_3 \setminus Y_2$ is non-empty, and every vertex in X'_2 is adjacent in G to every vertex in Y_2 , and every vertex in Y'_2 is adjacent in G to every vertex in X_2 . If exactly one of the sets X'_1 and Y'_1 is non-empty, say Y'_1 , then X'_2 is also non-empty, as otherwise $G[X_3, Y_3]$ would be disconnected. Hence, any vertices $x' \in X'_2$ and $y' \in Y'_1$ can augment any half graph $H_k^{\circ\circ}$ in $G[X_1, Y_1]$ into a half graph $H_{k+1}^{\circ\circ}$. Consequently, $\text{ch}(G[X_1, Y_1]) < \text{ch}(G[\{x'\} \cup X_1, \{y'\} \cup Y_1]) \leq \text{ch}(G_3)$. If both sets X'_1 and Y'_1 are non-empty, the argument is similar and we omit the details. \square

Theorem 5.49. *Let \mathcal{F} be a stable class of P_7 -free bipartite graphs. Then \mathcal{F} admits a constant-size equality-based adjacency labeling scheme, and hence $\text{SK}(\mathcal{F}_n) = O(1)$.*

Proof. Since \mathcal{F} is stable, it does not contain $\mathcal{C}^{\circ\circ}$ as a subclass. Let c be the maximum number such that $H_c^{\circ\circ} \in \mathcal{F}$, and let $G = (X, Y, E)$ be an arbitrary graph from \mathcal{F} .

By the above discussion G admits a $(\mathcal{Q}, 2(c+2))$ -decomposition tree, where \mathcal{Q} is the class consisting of bicliques and co-bicliques, and every P -node $G' = G[X', Y']$ is specified by the partition X'_1, \dots, X'_p of X' and the partition Y'_1, \dots, Y'_q of Y' as in Lemma 5.47, where p and q are bounded from above by $2(c+2)$.

We claim that the depth of such a decomposition tree is at most $6c$. To show this we associate with every node G' the pair $(\text{ch}(G'), \text{ch}(\overline{G}'))$ and we will prove that if the length of the path from the root G to a node G' is at least $6c$, then $\text{ch}(G') \leq 1$ or $\text{ch}(\overline{G}') \leq 1$, which means that G' is either a biclique or a co-biclique, and therefore is a leaf node.

Let \mathcal{P} be the path from the root to the node G' . By Claim 5.48 (1), among any three consecutive nodes of the path, there exists a P -node, or a pair of nodes labeled with \overline{D} and D respectively such that the \overline{D} -node is the parent of the D -node. In the former case, by Lemma 5.47, for the child node H' of the P -node H on the path \mathcal{P} , we have $(\text{ch}(H'), \text{ch}(\overline{H}')) \prec (\text{ch}(H), \text{ch}(\overline{H}))$. In the latter case, by Claim 5.48 (2), the child of the D -node on the path \mathcal{P} has the chain number strictly less than that of the D -node. In other words, for every node H in the path \mathcal{P} and its ancestor H' at distance 3 from H , we have $(\text{ch}(H'), \text{ch}(\overline{H}')) \prec (\text{ch}(H), \text{ch}(\overline{H}))$.

Now, since for the root node G we have $(\text{ch}(G), \text{ch}(\overline{G})) \prec (c, c+1)$, if \mathcal{P} has length at least $6c$, then $\text{ch}(G') \leq 1$ or $\text{ch}(\overline{G}') \leq 1$, as required. The result now follows from Lemma 5.25 and a simple observation that class \mathcal{Q} admits a constant-size equality-based adjacency labeling scheme. \square

6 Question III. Equality is Not Complete

Recall the notion of reductions between communication problems from Section 3.3. Many questions about constant-cost randomized communication would be answered if one could identify a *complete* problem for this class of problems under these reductions:

Definition 6.1. A communication problem \mathcal{P} is *complete* for the class of constant-cost randomized communication if every constant-cost randomized problem \mathcal{A} reduces to \mathcal{P} .

The most natural candidate for a complete problem is the EQUALITY problem. Note that all of our results in this paper so far, except for Cartesian products, have used reductions to EQUALITY. In this section we will prove that the EQUALITY problem is not complete for the class of constant-cost randomized communication. To this end we will show that 1-HAMMING DISTANCE does not reduce to EQUALITY.

Theorem 6.2. *1-HAMMING DISTANCE does not reduce to EQUALITY.*

This theorem was also proved independently and concurrently in [HHH23] with a very different Fourier-analytic argument.

In our proof, we will employ some results from the literature. We denote by H_d the d -dimensional hypercube, i.e., the d -wise Cartesian product $P_2^{\square d}$ of the single edge.

Theorem 6.3 ([ARSV06]). *For every k and $\ell \geq 6$, there exists $d_0(k, \ell)$ such that for every $d \geq d_0(k, \ell)$, every edge coloring of H_d with k colors contains a monochromatic induced cycle of length 2ℓ .*

For a graph G , its *equivalence covering number* $\text{eqc}(G)$ is the minimum number k such that there exist k equivalence graphs $F_i = (V, E_i), i \in [k]$, whose union $(V, \cup_{i=1}^k E_i)$ coincides with G . Similarly, for a bipartite graph $G = (U, W, E)$, its *bipartite equivalence covering number* $\text{beqc}(G)$ is the minimum number k such that there exist k bipartite equivalence graphs $F_i = (U, W, E_i), i \in [k]$, whose union $(U, W, \cup_{i=1}^k E_i)$ coincides with G .

Theorem 6.4 ([LNP80, Alo86]). *For every $n \geq 3$, it holds that $\text{eqc}(\overline{C_n}) \geq \log n - 1$.*

Corollary 6.5. *For every even $n \geq 4$, it holds that $\text{beqc}(\overline{C_n}) \geq \log n - 2$.*

Proof. Let $G = (U, W, E)$ be a bipartite graph isomorphic to $\overline{C_n}$ for some $n \geq 4$. Suppose towards the contradiction that there exist bipartite equivalence graphs $F_i = (U, W, E_i), i \in [k]$, whose union $(U, W, \cup_{i=1}^k E_i)$ coincides with G and $k < \log n - 2$. For each $i \in [k]$, let S_i be the (non-bipartite) equivalence graph obtained from F_i by turning each of its bicliques into a complete graph (this is done by adding edges between vertices of the same biclique that are in the same part). Also let $S = (U \cup W, E')$ be the equivalence graph with exactly two cliques, U and W . Then the union $S \cup (\cup_{i \in [k]} S_i)$ coincides with $\overline{C_n}$, implying $\text{eqc}(\overline{C_n}) \leq k + 1 < \log n - 1$, which is in contradiction with [Theorem 6.4](#). \square

For two binary vectors $x, y \in \{0, 1\}^t$, we write $x \preceq y$ if $x_i \leq y_i$ for all $i \in [t]$, and we write $x \prec y$ if $x \preceq y$ and $x \neq y$.

We now prove [Theorem 6.2](#).

Proof of Theorem 6.2. Let $\mathcal{A} = (A_n)_{n \in \mathbb{N}}$ and $\mathcal{B} = (B_n)_{n \in \mathbb{N}}$ be the 1-HAMMING DISTANCE problem and the EQUALITY problem respectively. More specifically, A_n is a $2^n \times 2^n$ matrix, where $A_n(i, j) = 1$ if and only if the binary representations of i and j differ on exactly one bit; and B_n is an $n \times n$ matrix, where $B_n(i, j) = 1$ if and only if $i = j$, i.e., B_n is the identity matrix.

By [Proposition 3.18](#), in order to prove the statement, we will show that \mathcal{A} does not matrix-reduce to $\text{QS}(\mathcal{B})$. To do this, we interpret the matrices of \mathcal{A} and \mathcal{B} as bipartite adjacency matrices of bipartite graphs. In particular, A_n corresponds to the disjoint union of two hypercube graphs of dimension n , and B_n corresponds to the matching graph, i.e., the graph in which every vertex has degree exactly 1. The set $\text{QS}(\mathcal{B})$ then corresponds to bipartite adjacency matrices of bipartite equivalence graphs.

Thus, we want to show that there does not exist a constant t such that every matrix $A \in \mathcal{A}$ can be expressed as $A = h(M_1, M_2, \dots, M_t)$, where $M_1, M_2, \dots, M_t \in \text{QS}(\mathcal{B})$ are bipartite adjacency matrices of some bipartite equivalence graphs and $h : \{0, 1\}^t \rightarrow \{0, 1\}$ is some Boolean function.

Suppose towards the contradiction that such a t exists. Let $k = 2^t$, $\ell = 2^{t+2}$, and $d \geq d_0(k, \ell)$, where $d_0(k, \ell)$ is the function from [Theorem 6.3](#). Since A_d , the bipartite adjacency matrix of two disjoint d -dimensional hypercube graphs, can be expressed as above via t matrices from $\text{QS}(\mathcal{B})$, the same is true for the bipartite adjacency matrix of a single d -dimensional hypercube graph.

Let $H_d = (U, W, E)$ be the hypercube graph of dimension d , and let $h : \{0, 1\}^t \rightarrow \{0, 1\}$ and $M_1, M_2, \dots, M_t \in \text{QS}(\mathcal{B})$ be such that $H_d(u, w) = h(M_1(u, w), M_2(u, w), \dots, M_t(u, w))$ for every $u \in U, w \in W$.

Define $\kappa : U \times W \rightarrow \{0, 1\}^t$ as $\kappa(u, w)_s = M_s(u, w)$ for every $s \in [t], u \in U$, and $w \in W$. Color every edge (u, w) of H_d with $\kappa(u, w)$. Since the edges of H_d are colored in at most $k = 2^t$ different colors, by [Theorem 6.3](#), it contains a monochromatic induced cycle $C = (U', W', E')$ of length $2\ell = 2^{t+3}$. Let $\kappa^* \in \{0, 1\}^t$ be the color of the edges of C .

Claim 1. *For all $u \in U'$ and $w \in W'$ that are not adjacent in C , we have $\kappa^* \prec \kappa(u, w)$.*

Proof. Since every connected component of a bipartite equivalence graph is a biclique, it follows that for every $i \in [t]$, $\kappa_i^* = 1$ implies $\kappa(u, w)_i = 1$ for all $u \in U'$ and $w \in W'$. Hence, $\kappa^* \preceq \kappa(u, w)$. Furthermore, if u and w are not adjacent in C , we have $\kappa(u, w) \neq \kappa^*$, as otherwise we would have $h(\kappa(u, w)) = h(\kappa^*) = 1$ and hence u and w would be adjacent. \square

Let now $I \subseteq [t]$ be the index set such that $i \in I$ if and only if $\kappa_i^* = 0$ and there exist $u \in U'$ and $w \in W'$ with $\kappa(u, w)_i = 1$. For every $i \in I$, let $F_i = (U', W', E'_i)$, where $E'_i = \{(u, w) \mid u \in U', w \in W', \kappa(u, w)_i = 1\}$. Note that each F_i is an induced subgraph of the bipartite equivalence graph defined by the bipartite adjacency matrix M_i , and thus it is itself a bipartite equivalence graph. By construction and Claim 1, we have that the union $\cup_{i \in I} F_i$ contains none of the edges of C and contains all non-edges between the sets U' and W' , i.e., the union coincides with \overline{C} . This implies that $\text{beqc}(\overline{C}) \leq |I| \leq t$. However, by [Corollary 6.5](#), $\text{beqc}(\overline{C}) \geq \log 2\ell - 2 \geq t + 1$, a contradiction. \square

Corollary 6.6. *EQUALITY is not complete for constant-cost randomized communication.*

Corollary 6.7. *There is no equality-based labeling scheme for the class of induced subgraphs of the hypercube.*

Acknowledgments

We thank Eric Blais for helpful discussions about this work, and we thank Abhinav Bom-mireddi, Renato Ferreira Pinto Jr., Sharat Ibrahimpur, and Cameron Seth for their comments on the presentation of this article. We thank the anonymous reviewers for their comments.

References

- [ACFL16] Aistis Atminas, Andrew Collins, Jan Foniok, and Vadim V Lozin. Deciding the bell number for hereditary graph properties. *SIAM Journal on Discrete Mathematics*, 30(2):1015–1031, 2016.
- [ACHS23] Manasseh Ahmed, Tsun-Ming Cheung, Hamed Hatami, and Kusha Sareen. Communication complexity of half-plane membership. Technical Report TR23-50, Electronic Colloquium on Computational Complexity (ECCC), 2023.
- [ACLZ15] Aistis Atminas, Andrew Collins, Vadim Lozin, and Victor Zamaraev. Implicit representations and factorial properties of graphs. *Discrete Mathematics*, 338(2):164–179, 2015.
- [AHKO22] Jung-ho Ahn, Kevin Hendrey, Donggyu Kim, and Sang-il Oum. Bounds for the twin-width of graphs. *SIAM Journal on Discrete Mathematics*, 36(3):2352–2366, 2022.

- [Ale92] Vladimir Evgen'evich Alekseev. Range of values of entropy of hereditary classes of graphs. *Diskretnaya Matematika*, 4(2):148–157, 1992.
- [Ale97] Vladimir Evgen'evich Alekseev. On lower layers of a lattice of hereditary classes of graphs. *Diskretnyi Analiz i Issledovanie Operatsii*, 4(1):3–12, 1997.
- [All09] Peter Allen. Forbidden induced bipartite graphs. *Journal of Graph Theory*, 60(3):219–241, 2009.
- [Alo86] Noga Alon. Covering graphs by the minimum number of equivalence relations. *Combinatorica*, 6(3):201–206, 1986.
- [ARSV06] Noga Alon, Radoš Radoičić, Benny Sudakov, and Jan Vondrák. A ramsey-type result for the hypercube. *Journal of Graph Theory*, 53(3):196–208, 2006.
- [ATYY17] Anurag Anshu, Dave Touchette, Penghui Yao, and Nengkun Yu. Exponential separation of quantum communication and classical information. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 277–288, 2017.
- [BBW00] József Balogh, Béla Bollobás, and David Weinreich. The speed of hereditary properties of graphs. *Journal of Combinatorial Theory, Series B*, 79(2):131–156, 2000.
- [BBW05] József Balogh, Béla Bollobás, and David Weinreich. A jump to the bell number for hereditary graph properties. *Journal of Combinatorial Theory, Series B*, 95(1):29–48, 2005.
- [BD02] Harry Buhrman and Ronald De Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002.
- [BD23] Édouard Bonnet and Hugues Déprés. Twin-width can be exponential in treewidth. *Journal of Combinatorial Theory, Series B*, 161:1–14, 2023.
- [BDS⁺24a] Édouard Bonnet, Julien Duron, John Sylvester, Viktor Zamaraev, and Maksim Zhukovskii. Small but unwieldy: A lower bound on adjacency labels for small classes. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1147–1165. SIAM, 2024.
- [BDS⁺24b] Édouard Bonnet, Julien Duron, John Sylvester, Viktor Zamaraev, and Maksim Zhukovskii. Tight bounds on adjacency labels for monotone graph classes. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2024.
- [BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 337–347. IEEE, 1986.
- [BGK⁺21] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1977–1996. SIAM, 2021.
- [BH21] Jakub Balabán and Petr Hliněný. Twin-width is linear in the poset width. In *16th International Symposium on Parameterized and Exact Computation (IPEC 2021)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2021.

- [BKTW20] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width: tractable for model checking. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 601–612. IEEE, 2020.
- [BT95] Béla Bollobás and Andrew Thomason. Projections of bodies and hereditary properties of hypergraphs. *Bulletin of the London Mathematical Society*, 27(5):417–424, 1995.
- [CER93] Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Handle-rewriting hypergraph grammars. *Journal of computer and system sciences*, 46(2):218–270, 1993.
- [Cha20] Amit Chakrabarti. One-way randomized communication complexity of greater-than. Theoretical Computer Science Stack Exchange, 2020. URL: <https://csttheory.stackexchange.com/q/48110> (version: 2020-12-30).
- [Cha23] Maurice Chandoo. Logical labeling schemes. *Discrete Mathematics*, 346(10):113565, 2023.
- [CHHS23] Tsun-Ming Cheung, Hamed Hatami, Kaave Hosseini, and Morgan Shirley. Separation of the factorization norm and randomized communication complexity. In *Proceedings of the Computational Complexity Conference (CCC 2023)*, 2023. Available at ECCC, TR22-165.
- [CLR20] Victor Chepoi, Arnaud Labourel, and Sébastien Ratel. On density of subgraphs of Cartesian products. *Journal of Graph Theory*, 93(1):64–87, 2020.
- [CLV19] Arkadev Chattopadhyay, Shachar Lovett, and Marc Vinyals. Equality alone does not simulate randomness. In *34th Computational Complexity Conference (CCC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.
- [CS18] Artem Chernikov and Sergei Starchenko. A note on the Erdős-Hajnal property for stable graphs. *Proceedings of the American Mathematical Society*, 146(2):785–790, 2018.
- [DEG⁺21] Vida Dujmović, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin. Adjacency labelling for planar graphs (and beyond). *Journal of the ACM (JACM)*, 68(6):1–33, 2021.
- [EHK22] Louis Esperet, Nathaniel Harms, and Andrey Kupavskii. Sketching distances in monotone graph classes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.
- [EHZ24] Louis Esperet, Nathaniel Harms, and Viktor Zamaraev. Optimal Adjacency Labels for Subgraphs of Cartesian Products. *SIAM Journal on Discrete Mathematics*, 2024.
- [ES35] Paul Erdős and George Szekeres. A combinatorial problem in geometry. *Compositio mathematica*, 2:463–470, 1935.
- [FGHH24] Yuting Fang, Mika Göös, Nathaniel Harms, and Pooya Hatami. Constant-cost communication does not reduce to k -Hamming distance. *arXiv preprint arXiv:2407.20204*, 2024.
- [FHHH24] Yuting Fang, Lianna Hambardzumyan, Nathaniel Harms, and Pooya Hatami. No complete problem for constant-cost randomized communication. In *Proceedings of the Symposium on Theory of Computing (STOC 2024)*, 2024.

- [Fit19] Matthew Fitch. Implicit representation conjecture for semi-algebraic graphs. *Discrete Applied Mathematics*, 259:53–62, 2019.
- [FK09] Pierre Fraigniaud and Amos Korman. On randomized representations of graphs using short labels. In *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures - SPAA 2009*. ACM Press, 2009.
- [GPT22] Jakub Gajarský, Michał Pilipczuk, and Szymon Toruńczyk. Stable graphs of bounded twin-width. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–12, 2022.
- [GPW18] Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *Computational Complexity*, 27(2):245–304, 2018.
- [Har20] Nathaniel Harms. Universal communication, universal graphs, and graph labeling. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- [HH22] Hamed Hatami and Pooya Hatami. The implicit graph conjecture is false. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1134–1137. IEEE, 2022.
- [HHH22] Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. A counter-example to the probabilistic universal graph conjecture via randomized communication complexity. *Discrete Applied Mathematics*, 322:117–122, 2022.
- [HHH23] Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. Dimension-free bounds and structural results in communication complexity. *Israel Journal of Mathematics*, 253(2):555–616, 2023.
- [HHM23] Hamed Hatami, Kaave Hosseini, and Xiang Meng. A Borsuk-Ulam lower bound for sign-rank and its applications. In *Proceedings of the 55th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2023)*, 2023.
- [HHP⁺22] Hamed Hatami, Pooya Hatami, William Pires, Ran Tao, and Rosie Zhao. Lower bound methods for sign-rank and their limitations. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.
- [HIK11] Richard Hammack, Wilfried Imrich, and Sandi Klavžar. *Handbook of product graphs*. CRC press, 2011.
- [HSZZ06] Wei Huang, Yaoyun Shi, Shengyu Zhang, and Yufan Zhu. The communication complexity of the hamming distance problem. *Information Processing Letters*, 99(4):149–153, 2006.
- [HWZ22] Nathaniel Harms, Sebastian Wild, and Viktor Zamaraev. Randomized communication and implicit graph representations. In *Proceedings of the Symposium on Theory of Computing (STOC 2022)*, 2022.
- [HZ24] Nathaniel Harms and Viktor Zamaraev. Randomized communication and implicit representations for matrices and graphs of small sign-rank. In *Proceedings of the 2024*

- Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1810–1833. SIAM, 2024.
- [KL15] Sergey Kitaev and Vadim Lozin. *Words and graphs*. Springer, 2015.
- [KM12] Ross J Kang and Tobias Müller. Sphere and dot product representations of graphs. *Discrete & Computational Geometry*, 47(3):548–568, 2012.
- [KNR92] Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. *SIAM Journal on Discrete Mathematics*, 5(4):596–603, 1992.
- [KNR99] Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.
- [LNP80] László Lovász, J Nešetřil, and Ales Pultr. On a product dimension of graphs. *Journal of Combinatorial Theory, Series B*, 29(1):47–67, 1980.
- [LV08] VV Lozin and J Volz. The clique-width of bipartite graphs in monogenic classes. *International Journal of Foundations of Computer Science*, 19(2):477–494, 2008.
- [LZ17] Vadim Lozin and Viktor Zamaraev. The structure and the number of P_7 -free bipartite graphs. *European Journal of Combinatorics*, 65:143–153, 2017.
- [Mad67] Wolfgang Mader. Homomorphieeigenschaften und mittlere kantendichte von graphen. *Mathematische Annalen*, 174(4):265–268, 1967.
- [MNSW98] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.
- [MS14] Maryanthe Malliaris and Saharon Shelah. Regularity lemmas for stable graphs. *Transactions of the American Mathematical Society*, 366(3):1551–1585, 2014.
- [Mul89] John H Muller. Local structure in graph classes. 1989.
- [New91] Ilan Newman. Private vs. common random bits in communication complexity. *Information processing letters*, 39(2):67–71, 1991.
- [Nik20] Sasho Nikolov. One-way randomized communication complexity of greater-than. Theoretical Computer Science Stack Exchange, 2020. URL: <https://cstheory.stackexchange.com/q/48108> (version: 2020-12-29).
- [Nis93] Noam Nisan. The communication complexity of threshold gates. *Combinatorics, Paul Erdos is Eighty*, 1:301–315, 1993.
- [NK96] Noam Nisan and Eyal Kushilevitz. *Communication Complexity*. Cambridge University Press, 1996.
- [NMP⁺21] Jaroslav Nešetřil, Patrice Ossona de Mendez, Michał Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. Rankwidth meets stability. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2014–2033. SIAM, 2021.
- [NP24] Moni Naor and Eugene Pikel. Adjacency sketches in adversarial environments. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*, pages 1067–1098. SIAM, 2024.

- [PS86] Ramamohan Paturi and Janos Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986.
- [PSW20] Toniann Pitassi, Morgan Shirley, and Thomas Watson. Nondeterministic and randomized boolean hierarchies in communication complexity. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- [Rad64] Richard Rado. Universal graphs and universal functions. *Acta Arithmetica*, 4(9):331–340, 1964.
- [RS86] Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.
- [RS15] Sivaramakrishnan Natarajan Ramamoorthy and Makrand Sinha. On the communication complexity of greater-than. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 442–444. IEEE, 2015.
- [RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity and Applications*. Cambridge University Press, 2020.
- [Sağ18] Mert Sağlam. Near log-convexity of measured heat in (discrete) time and consequences. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 967–978. IEEE, 2018.
- [Sch99] Edward R Scheinerman. Local representations using very short labels. *Discrete mathematics*, 203(1-3):287–290, 1999.
- [Smi88] D. V. Smirnov. Shannon’s information methods for lower bounds for probabilistic communication complexity. Master’s thesis, Moscow University, 1988.
- [Spi03] Jeremy P Spinrad. *Efficient graph representations*. American Mathematical Society, 2003.
- [SS22] André Schidler and Stefan Szeider. A sat approach to twin-width. In *2022 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 67–77. SIAM, 2022.
- [SZ94] Edward R Scheinerman and Jennifer Zito. On the size of hereditary classes of graphs. *Journal of Combinatorial Theory, Series B*, 61(1):16–39, 1994.
- [Vio15] Emanuele Viola. The communication complexity of addition. *Combinatorica*, 35(6):703–747, 2015.
- [Yao03] Andrew Chi-Chih Yao. On the power of quantum fingerprinting. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 77–81, 2003.

A Missing Proofs from Section 3

A.1 Probability Boosting and Derandomization

Proposition 3.4. *Let \mathcal{F} be a class of graphs. For any $\delta \in (0, 1/2)$, there is an adjacency sketch with error δ and size at most $O(\text{SK}(\mathcal{F}_n) \cdot \log \frac{1}{\delta})$. Equivalently, if there is a PUG $U = (U_n)_{n \in \mathbb{N}}$ for \mathcal{F} with size $|U_n|$, then there is a PUG $U' = (U'_n)_{n \in \mathbb{N}}$ for \mathcal{F} with error δ and $|U'_n| \leq |U_n|^{O(\log(1/\delta))}$.*

Proof. Write $s(n) := \text{SK}(\mathcal{F}_n)$ and let $G \in \mathcal{F}_n$. Then there is a distribution \mathcal{S} over functions $\text{sk} : V(G) \rightarrow \{0, 1\}^{s(n)}$ and a decoder $D : \{0, 1\}^{s(n) \times s(n)} \rightarrow \{0, 1\}$ satisfying the definition of an adjacency sketch. Consider the following adjacency sketch. Sample $\text{sk}_1, \dots, \text{sk}_k$ independently from \mathcal{S} . To each vertex $x \in V(G)$ assign the label $\text{sk}'(x) = (\text{sk}_1(x), \dots, \text{sk}_k(x))$. On input $(\text{sk}'(x), \text{sk}'(y))$, the decoder will output $\text{majority}(D(\text{sk}_1(x), \text{sk}_1(y)), \dots, D(\text{sk}_k(x), \text{sk}_k(y)))$.

For each $i \in [k]$, let $X_i = 1$ if $D(\text{sk}_i(x), \text{sk}_i(y)) = \mathbf{1}[(x, y) \in E(G)]$ and $X_i = 0$ otherwise; and let $X = \sum_{i=1}^k X_i$. Observe that for each $i \in [k]$, $\mathbb{E}[X_i] \geq 2/3$, so $\mathbb{E}[X] \geq 3k/2$. Then, by the Chernoff bound, the probability that the decoder fails is

$$\mathbb{P}[X \leq k/2] \leq \mathbb{P}[X \leq \mathbb{E}[X]/3] \leq e^{-\frac{k}{3}}.$$

This is at most δ when $k = 3 \ln(1/\delta)$. □

Proposition 3.5. *For any hereditary graph class \mathcal{F} , there is an adjacency labeling scheme of size $O(\text{SK}(\mathcal{F}) \cdot \log n)$. If there is a randomized algorithm which produces the adjacency sketch in time $\text{poly}(n)$, then there is a randomized algorithm which produces the adjacency labels in expected time $\text{poly}(n)$.*

Proof. Let $G \in \mathcal{F}_n$. Using [Proposition 3.4](#) for $\delta = 1/n^2$, we obtain an adjacency sketch with error probability δ and size $c(n) = O(\text{SK}(\mathcal{F}_n) \cdot \log(1/\delta)) = O(\text{SK}(\mathcal{F}_n) \cdot \log n)$. For a fixed sketch function $\text{sk} : V(G) \rightarrow \{0, 1\}^{c(n)}$, write Δ_{sk} for the number of pairs x, y such that the decoder outputs the incorrect value on input $\text{sk}(x), \text{sk}(y)$. Then, by the union bound, for a randomly chosen sketch function sk ,

$$\mathbb{E}[\Delta_{\text{sk}}] \leq \delta \binom{n}{2} \leq 1/2n.$$

Then there exists a fixed sk with $\Delta_{\text{sk}} \leq 1/2n < 1$ so $\Delta_{\text{sk}} = 0$; this sk is an adjacency labeling scheme for G . Furthermore, for $\mu = \mathbb{E}[\Delta_{\text{sk}}]$, Markov's inequality gives

$$\mathbb{P}[\Delta_{\text{sk}} > 0] = \mathbb{P}[\Delta_{\text{sk}} \geq 1] = \mathbb{P}[\Delta_{\text{sk}} \geq 2n\mu] \leq \frac{\mu}{n\mu} = 1/2n,$$

so a random sk function is a (deterministic) adjacency labeling scheme for G with probability at least $1/2$. Therefore, if there is a randomized algorithm producing the adjacency sketch in time $\text{poly}(n)$, then from the proof of [Proposition 3.4](#) we see that we can produce the random sk function with a randomized algorithm in time $\text{poly}(n)$; in expectation we must sample 2 instances of sk to find one with $\Delta_{\text{sk}} = 0$. It takes time at most $\text{poly}(n)$ to check that all the labels are correct, since we must check n^2 pairs of vertices. □

A.2 Lower Bound from Greater-Than

A.2.1 Proof of the lower bound

For the proof of the next statement, let us briefly describe the message passing (SMP) model of communication. In this model, given (private) inputs $x, y \in [n]$ to problem $f_n : [n] \times [n] \rightarrow \{0, 1\}$, Alice and Bob use shared randomness to send random messages $A(x), B(y)$ to a third-party referee, who must output $f_n(x, y)$ with probability at least $2/3$ over the choice of messages. The complexity of the protocol is $\max_{x, y} \max(|A(x)|, |B(y)|)$. It is known that on domain $[n]$, the SMP complexity of GREATER-THAN is $\Theta(\log n)$, in contrast to the two-way randomized communication complexity, which is $\Theta(\log \log n)$, where the upper bound is due to [\[Nis93\]](#) (see discussion in [Appendix A.2.2](#)).

Proposition 3.11. *If a hereditary graph class \mathcal{F} is not stable, then $\text{SK}(\mathcal{F}) = \Omega(\log n)$.*

Proof. This follows from the fact that an adjacency sketch for \mathcal{F} can be used to construct a communication protocol for GREATER-THAN in the public-coin SMP model of communication. The construction is as follows. Let D be the decoder for an adjacency sketch for \mathcal{F} . Given inputs $x, y \in [n]$, Alice and Bob can compute $\text{GT}_n(x, y)$ in the SMP model by choosing a graph $G \in \mathcal{F}$ with $\text{ch}(G) = n$, so there exist disjoint sets of vertices $\{a_1, \dots, a_n\}, \{b_1, \dots, b_n\}$ such that (a_i, b_j) are adjacent if and only if $i \leq j$. Since \mathcal{F} is hereditary, the induced subgraph $H \sqsubset G$ on vertices $\{a_1, \dots, a_n, b_1, \dots, b_n\}$ is in \mathcal{F} . Alice and Bob draw random sketches $\text{sk}(a_x), \text{sk}(b_y)$ according to the adjacency sketch for H , and send them to the referee, who outputs $D(\text{sk}(a_x), \text{sk}(b_y))$. This communication protocol has complexity at most $\text{SK}(\mathcal{F})$, so by the lower bound on the SMP complexity of GREATER-THAN, we must have $\text{SK}(\mathcal{F}) = \Omega(\log n)$. \square

A.2.2 Bibliographic remark on Greater-Than

Recall the lower bound for Greater-Than:

Theorem A.1. *Any public-coin randomized SMP communication protocol for GREATER-THAN on domain $[n]$ requires $\Omega(\log n)$ bits of communication.*

Lower bounds for the GREATER-THAN problem in various models appear in [KNR99, MNSW98, Vio15, RS15, ATYY17]. The above theorem is stated in [KNR99] and [MNSW98]; in the latter it is also credited to [Smi88]. In [KNR99] the theorem is stated for one-way *private-coin* communication; the result for public-coin SMP communication follows from the fact that public-coin protocols for problems with domain size n can save at most $O(\log \log n)$ bits of communication over the private-coin protocol due to Newman’s theorem [New91]. We remark that the tight lower bound of $\Omega(\log \log n)$ for GREATER-THAN in the two-way communication model follows from Theorem A.1 since there is at most an exponential gap between SMP and two-way communication.

However, as noted in a CSTheory StackExchange question of Nikolov [Nik20], the complete proof is not provided in either of [KNR99, MNSW98]. The same lower bound for *quantum* communication complexity is proved in [ATYY17], which implies the above result. A direct proof for classical communication complexity was suggested as an answer to [Nik20] by Chakrabarti [Cha20]; we state this direct proof here for completeness and we thank Eric Blais for communicating this reference to us. We require the AUGMENTED-INDEX communication problem and its lower bound from [MNSW98].

Definition A.2 (Augmented-Index). In the AUGMENTED-INDEX communication problem, Alice receives input $x \in \{0, 1\}^k$ and Bob receives an integer $i \in [k]$ along with the values x_j for all $j > i$. Bob should output the value x_i .

Theorem A.3 ([MNSW98]). *Any public-coin randomized one-way communication protocol for AUGMENTED-INDEX requires $\Omega(k)$ bits of communication.*

Proof of Theorem A.1, [Cha20]. Given inputs $x \in \{0, 1\}^k$ and $i \in [k]$ to the AUGMENTED INDEX problem, Bob constructs the string $y \in \{0, 1\}^k$ where $y_j = x_j$ for all $j > i$ and $y_i = 0$, and $y_j = 1$ for all $j < i$. Consider the numbers $a, b \in [2^k]$ where the binary representation of a is x , with bit k being the most significant and bit 1 the least significant, and the binary representation of b is y , with the bits in the same order. If $x_i = 1$, then since $y_i = 0$ and $y_j = x_j$ for $j > i$, it holds that $b < a$. If $x_i = 0$, then since $y_j = x_j$ for $j \geq i$ and $y_j = 1$ for $j < i$ it holds that $b \geq a$. Therefore, computing GREATER-THAN on inputs a, b will solve AUGMENTED INDEX. By Theorem A.3, the communication cost of GREATER-THAN for $n = 2^k$ is at least $\Omega(k) = \Omega(\log n)$. \square

A.3 Missing Proof for Equality-Based Labeling

Proposition 3.26. *The following are equivalent for a hereditary graph class \mathcal{F} :*

1. \mathcal{F} admits a constant-size equality-based labeling scheme;
2. The graph class \mathcal{F} reduces to the class of equivalence graphs;
3. The communication problem $\text{ADJ}_{\mathcal{F}}$ reduces to EQUALITY.

Therefore, if \mathcal{F} admits a constant-size equality-based labeling scheme, it admits a constant-size adjacency sketch (and hence a constant-size PUG).

Proof. We prove the equivalence between items (1) and (3). The equivalence between items (2) and (3) holds essentially by definition. Suppose \mathcal{F} admits an (s, k) -equality based labeling scheme where s, k are constants. For any matrix $M \in \text{ADJ}_{\mathcal{F}}$, which is the adjacency matrix of $G_M \in \mathcal{F}$, two players with inputs x, y can use s calls to the EQUALITY oracle to send the prefix $p(x) \in \{0, 1\}^s$ to the other player. They may then use k^2 calls to the EQUALITY oracle to compute the entries of the matrix $Q_{x,y}$, from which they can compute the output $D_{p(x),p(y)}(Q_{x,y})$ of the decoder.

Now, suppose $\text{ADJ}_{\mathcal{F}}$ reduces to EQUALITY. By Proposition 3.18, $\text{ADJ}_{\mathcal{F}}$ matrix-reduces to QS(EQUALITY). Notices that QS(EQUALITY) is the set of bipartite adjacency matrices of bipartite equivalence graphs, and any bipartite equivalence graph $G = (X, Y, E)$ can be defined by a pair of functions $a : X \rightarrow [k]$ and $b : Y \rightarrow [k]$ such that for every $x \in X$ and $y \in Y$, x and y are adjacent in G if and only if $\mathbb{1}[a(x) = b(y)] = 1$. This implies that for every $n_1 \times n_2$ matrix $M \in \text{ADJ}_{\mathcal{F}}$ there is a function $h : [k]^k \rightarrow \{0, 1\}$ and maps $a_i : [n_1] \rightarrow [k]$, $b_i : [n_2] \rightarrow [k]$, $i \in [k]$, such that for any $x \in [n_1]$ and $y \in [n_2]$

$$M(x, y) = h(\mathbb{1}[a_1(x) = b_1(y)], \mathbb{1}[a_2(x) = b_2(y)], \dots, \mathbb{1}[a_k(x) = b_k(y)]). \quad (1)$$

We then obtain a constant-size equality-based labeling scheme for \mathcal{F} by assigning to each vertex x of $G_M \in \mathcal{F}$ with adjacency matrix $M \in \text{ADJ}_{\mathcal{F}}$ a label consisting of a prefix encoding the function h , and the equality codes $a_1(x), \dots, a_k(x), b_1(x), \dots, b_k(x)$, and defining the decoder as the function which computes Equation 1. \square

B The Lattice of Hereditary Graph Classes

We briefly survey some known facts about the lattice of hereditary graph classes, which may help to put our new questions about constant-size PUGs and constant-cost communication complexity into context. In Appendix B.1 we describe how the lattice is partitioned into “layers” based upon the speed of the class, and in Appendix B.2 we describe the “minimal” hereditary graph classes at different speeds, and how stability relates to these minimal classes.

The hereditary graph classes form a lattice, since for any two hereditary graph classes \mathcal{F} and \mathcal{H} , it holds that $\mathcal{F} \cap \mathcal{H}$ and $\mathcal{F} \cup \mathcal{H}$ are also hereditary graph classes. In this section we review the structure of this lattice, and give some basic results that place the set of constant-PUG classes within this lattice. For an illustrated summary of this section, see Figure 10.

B.1 The Speed of Hereditary Graph Families

The speed $|\mathcal{F}_n|$ of a hereditary graph class cannot be arbitrary. Classic results of Alekseev [Ale92, Ale97], Bollobás & Thomason [BT95], and Scheinerman & Zito [SZ94] have classified some of the possible speeds of hereditary graph classes. Scheinerman & Zito [SZ94] and Alekseev [Ale97] showed that the four smallest layers of hereditary graph classes are the following:

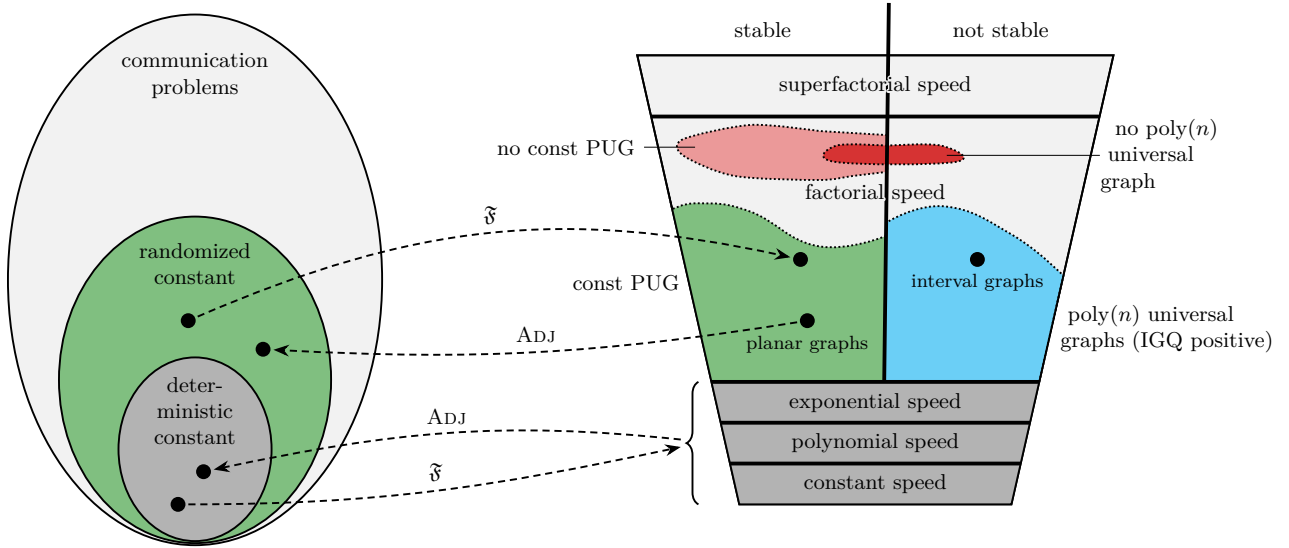


Figure 9: The relationship between layers of the lattice of hereditary graph classes (on the right), and communication complexity (on the left). The stability condition partitions the factorial layer. The red regions, indicating the existence of graph classes with no constant-size PUG and no $\text{poly}(n)$ -size universal graph, are due to [HH22, HHH22].

1. The *constant* layer contains classes \mathcal{F} with $\log |\mathcal{F}_n| = \Theta(1)$, and hence $|\mathcal{F}_n| = \Theta(1)$,
2. The *polynomial* layer contains classes \mathcal{F} with $\log |\mathcal{F}_n| = \Theta(\log n)$,
3. The *exponential* layer contains classes \mathcal{F} with $\log |\mathcal{F}_n| = \Theta(n)$,
4. The *factorial* layer contains classes \mathcal{F} with $\log |\mathcal{F}_n| = \Theta(n \log n)$.

The graph classes with *subfactorial* speed (the first three layers) have simple structure [SZ94, Ale97]. As demonstrated by earlier examples, the factorial layer is substantially richer and includes many graph classes of theoretical or practical importance. Despite this, no general characterization is known for them apart from the definition.

The first three slowest layers correspond exactly to constant-cost *deterministic* communication problems, under the equivalence between communication problems and hereditary graph classes described in Section 3.2. Scheinerman [Sch99] showed that a hereditary class \mathcal{F} admits a constant-size adjacency labeling scheme (i.e. a constant-cost deterministic communication protocol for $\text{ADJ}_{\mathcal{F}}$) if and only if it belongs to the *constant*, *polynomial*, or *exponential* layer. Such classes have a bounded number of equivalence classes of vertices, where two vertices x, y are equivalent if their neighborhoods satisfy $N(x) \setminus \{y\} = N(y) \setminus \{x\}$.

Proposition B.1. *A communication problem f admits a constant-cost deterministic protocol if and only if $\mathfrak{F}(f)$ is in the constant, polynomial, or exponential layer. A hereditary graph class \mathcal{F} is in the constant, polynomial, or exponential layer if and only if there is a constant-cost deterministic protocol for $\text{ADJ}_{\mathcal{F}}$.*

On the other hand, adjacency labels for a factorial class must have size $\Omega(\log n)$ since graphs in the minimal factorial classes can have $\Omega(n)$ equivalence classes of vertices, and each equivalence class requires a unique label. So there is a jump in label size from $O(1)$ (in the subfactorial layers) to $\Omega(\log n)$ (in the factorial layers), so that there is no hereditary graph class with label size between

$O(1)$ and $\Omega(\log n)$. The original version of this paper conjectured that a similar gap between $O(1)$ and $\Omega(\log n)$ should occur for adjacency *sketching* too; but this is not true, see [HHH22].

B.2 Minimal Factorial Families

The factorial layer has a set of 9 *minimal* classes, which satisfy the following:

1. Every factorial class \mathcal{F} contains at least one minimal class;
2. For each minimal class \mathcal{M} , any hereditary subclass $\mathcal{M}' \subset \mathcal{M}$ has subfactorial speed.

These classes were identified by Alekseev [Ale97], and similar results were independently obtained by Balogh, Bollobás, & Weinreich [BBW00]. Each minimal factorial class is either a class of bipartite graphs, or a class of *co-bipartite* graphs (i.e., complements of bipartite graphs), or a class of *split* graphs (i.e., graphs whose vertex set can be partitioned into a clique and an independent set). Six of the minimal classes are the following:

- $\mathcal{M}^{\circ\circ}$ is the class of bipartite graphs of degree at most 1.
- $\mathcal{M}^{\bullet\circ}$ is the class of graphs whose vertex set can be partitioned into a clique and an independent set such that every vertex in each of the parts is adjacent to at most one vertex in the other part.
- $\mathcal{M}^{\bullet\bullet}$ is the class of graphs whose vertex set can be partitioned into two cliques such that every vertex in each of the parts is adjacent to at most one vertex in the other part.
- $\mathcal{L}^{\circ\circ}, \mathcal{L}^{\bullet\circ}, \mathcal{L}^{\bullet\bullet}$ are defined similarly to the classes $\mathcal{M}^{\circ\circ}, \mathcal{M}^{\bullet\circ}, \mathcal{M}^{\bullet\bullet}$, respectively, with the difference that vertices in each of the parts are adjacent to all but at most one vertex in the other part.

The other three minimal classes motivate our focus on the *stable* factorial classes. They are defined as follows. For any $k \in \mathbb{N}$, recall that the *half-graph* (see Figure 2) is the bipartite graph $H_k^{\circ\circ}$ with vertex sets $\{a_1, \dots, a_k\}$ and $\{b_1, \dots, b_k\}$, where the edges are exactly the pairs (a_i, b_j) that satisfy $i \leq j$. The *threshold graph* $H_k^{\bullet\circ}$ is the graph defined the same way, except including all edges (a_i, a_j) where $i \neq j$. The *co-half-graph* $H_k^{\bullet\bullet}$ is the graph defined the same way as the threshold graph but also including all edges (b_i, b_j) for $i \neq j$. We define the following hereditary classes:

$$\mathcal{C}^{\circ\circ} := \text{cl}\{H_k^{\circ\circ} : k \in \mathbb{N}\}, \quad \mathcal{C}^{\bullet\circ} := \text{cl}\{H_k^{\bullet\circ} : k \in \mathbb{N}\}, \quad \mathcal{C}^{\bullet\bullet} := \text{cl}\{H_k^{\bullet\bullet} : k \in \mathbb{N}\}.$$

Proposition B.2 ([Ale97]). *The minimal factorial classes are*

$$\mathcal{M}^{\circ\circ}, \mathcal{M}^{\bullet\circ}, \mathcal{M}^{\bullet\bullet}, \mathcal{L}^{\circ\circ}, \mathcal{L}^{\bullet\circ}, \mathcal{L}^{\bullet\bullet}, \mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet}.$$

See Figure 10. It is clear from the definitions that the classes $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet}$ are not stable, while the other minimal classes are. The following statement is easily proved from Proposition 3.11.

Fact B.3. $\mathcal{M}^{\circ\circ}, \mathcal{M}^{\bullet\circ}, \mathcal{M}^{\bullet\bullet}, \mathcal{L}^{\circ\circ}, \mathcal{L}^{\bullet\circ}, \mathcal{L}^{\bullet\bullet}$ admit constant-size equality-based labeling schemes (and therefore constant-size PUGs), while $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet}$ have PUGs of size $n^{\Theta(1)}$.

A consequence of Ramsey's theorem is that a hereditary graph class \mathcal{F} is stable if and only if it does not include any of $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet}$:

Proposition B.4. *Let \mathcal{F} be a hereditary class of graphs. Then \mathcal{F} has bounded chain number if and only if $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet} \not\subseteq \mathcal{F}$.*

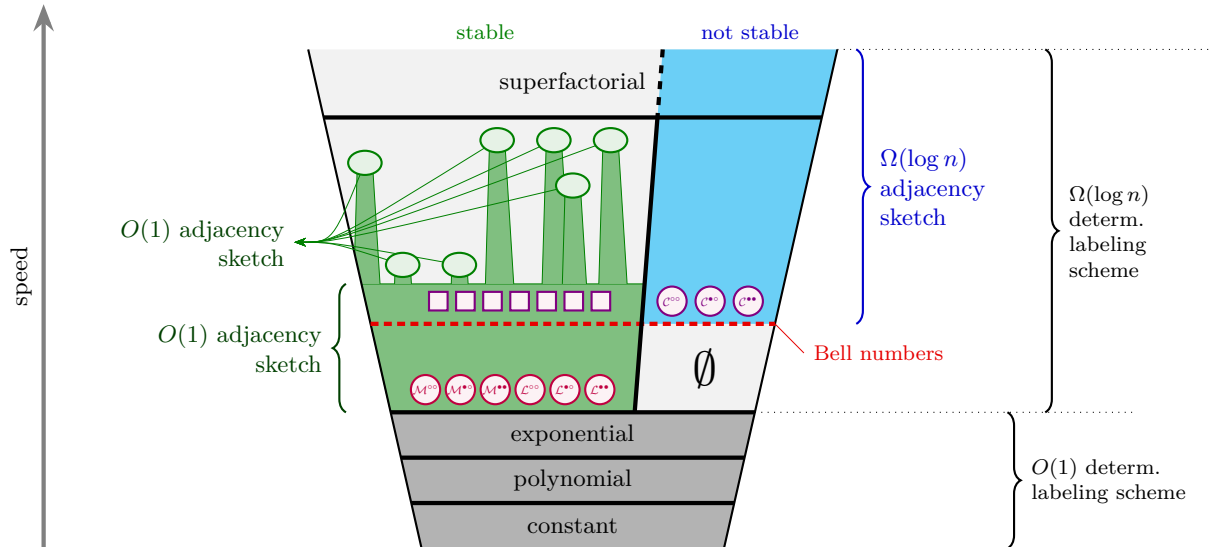


Figure 10: Overview of our results (in green) in relation to the lattice of hereditary graph classes. Circles are minimal factorial classes; purple shapes are minimal classes above the Bell numbers.

Unlike standard universal graphs, PUGs exhibit a large quantitative gap between the chain-like graphs and the other minimal factorial classes, suggesting that stable factorial classes behave much differently than other factorial classes and may be worth studying separately, which has not yet been done in the context of understanding the factorial layer of graph classes.

The Bell numbers threshold. There is another speed threshold within the factorial layer: the Bell numbers threshold. The Bell number B_n is the number of different set partitions of $[n]$, or equivalently the number of n -vertex equivalence graphs; asymptotically it is $B_n \sim (n/\log n)^n$. Similarly to the factorial layer itself, there is a set of *minimal* classes above the Bell numbers. However, unlike the factorial layer, the set of minimal classes above the Bell numbers is *infinite*, and it has been characterized explicitly [BBW05, ACFL16]. Once again, the classes $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet}$ are minimal. This means that *all* hereditary classes below the Bell numbers are stable. Structural properties of these classes were given in [BBW00], which can be used to prove the following.

Proposition B.5. *Let \mathcal{F} be a hereditary graph class. Then:*

1. *If \mathcal{F} is a minimal class above the Bell numbers, then \mathcal{F} admits a constant-size equality-based labeling scheme (and therefore a constant-size PUG), unless $\mathcal{F} \in \{\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet}\}$.*
2. *If \mathcal{F} has speed below the Bell numbers, then \mathcal{F} admits a constant-size equality-based labeling scheme (and therefore a constant-size PUG).*

The proof of this statement is routine, given the structural characterization of these graph classes in [BBW00, BBW05, ACFL16], and we omit it.