Randomized Communication and the Implicit Graph Conjecture

Nathaniel Harms*
University of Waterloo, Canada
nharms@uwaterloo.ca

Sebastian Wild
University of Liverpool, UK
sebastian.wild@liverpool.ac.uk

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

June 21, 2022

Abstract

The most basic lower-bound question in randomized communication complexity is: Does a given problem have constant cost, or non-constant cost? We observe that this question has a deep connection to implicit graph representations in structural graph theory. Specifically, constant-cost communication problems correspond to hereditary graph families that admit constant-size *adjacency sketches*, or equivalently constant-size *probabilistic universal graphs (PUGs)*, and these graph families are a subset of families that admit adjacency labeling schemes of size $O(\log n)$, which are the subject of the well-studied *implicit graph question (IGQ)*.

We initiate the study of the hereditary graph families that admit constant-size PUGs, with the two (equivalent) goals of (1) giving a structural characterization of randomized constant-cost communication problems, and (2) resolving a probabilistic version of the IGQ. For each family $\mathcal{F}$ studied in this paper (including the monogenic bipartite families, product graphs, interval and permutation graphs, families of bounded twin-width, and others), it holds that the subfamilies $\mathcal{H} \subseteq \mathcal{F}$ are either *stable* (in a sense relating to model theory), in which case they admit constant-size PUGs (i.e. adjacency sketches), or they are not stable, in which case they do not.

The correspondence between communication problems and hereditary graph families allows for a probabilistic method of constructing adjacency labeling schemes. By this method, we show that the induced subgraphs of any Cartesian products G^d are positive examples to the IGQ, also giving a bound on the number of unique induced subgraphs of any graph product. We prove that this probabilistic construction cannot be "naïvely derandomized" by using an EQUALITY oracle, implying that the EQUALITY oracle cannot simulate the k-HAMMING DISTANCE communication protocol.

As a consequence of our results, we obtain constant-size sketches for deciding $\mathsf{dist}(x, y) \leq k$ for vertices x, y in any stable graph family with bounded twin-width, answering an open question about planar graphs from [Har20]. This generalizes to constant-size sketches for deciding first-order formulas over the same graphs.

*Research supported by NSERC.

Contents

1 Introduction

In this paper we study a new connection between constant-cost randomized[1] communication and the long-standing Implicit Graph Conjecture (IGC) in structural graph theory [KNR92, Spi03], which have been studied independently for decades. Randomized communication is a central topic in communication complexity [GPW18, CLV19, PSW20, HHH21], and the most basic question is, *Does a given problem have a randomized protocol with only constant cost?* For example, the EQUALITY problem has constant cost [NK96], while the GREATER-THAN problem[2] has non-constant cost. What are the conditions for a communication problem to have a constant-cost protocol? We show that this question has deep connections to the IGC, which has motivated a large body of research in graph theory (e.g. [KM12, ACLZ15, BGK+21, DEG+21]), with limited progress towards settling it. Our question is in fact equivalent to a *probabilistic* version of the IGC (see Conjecture 1.2).

To state the IGC, we define the following. A hereditary graph family $\mathcal{F}$ is a set of (labeled) graphs, closed under isomorphism and under taking induced subgraphs. Write $\mathcal{F}_n$ for the set of graphs in $\mathcal{F}$ with vertex set $[n]$. A universal graph for $\mathcal{F}$ is a sequence (U_n) of graphs where U_n contains all $G \in \mathcal{F}_n$ as induced subgraphs; the function $n \mapsto |U_n|$ is the *size* of U. A universal graph of size $\mathrm{poly}(n)$ is equivalent to an $O(\log n)$-size *adjacency labeling scheme*, which assign labels to the vertices of $G \in \mathcal{F}$ such that adjacency between x and y can be determined from their labels without knowing G [KNR92]. The function $n \mapsto |\mathcal{F}_n|$ is called the *speed* of $\mathcal{F}$. The hereditary families with speed $2^{\Theta(n \log n)}$ are called *factorial* [SZ94, Ale97, BBW00]. The IGC is as follows:

Implicit Graph Conjecture. A hereditary graph family $\mathcal{F}$ admits a universal graph of size $\mathrm{poly}(n)$ (i.e. an $O(\log n)$-size adjacency labeling scheme) if and only if it has at most factorial speed.

The connection: We refer to hereditary graph families with universal graphs of size $\mathrm{poly}(n)$ as *positive examples to the IGC*. In Section 1.1 we show that constant-cost communication problems correspond exactly to a certain set of positive examples to the IGC. Briefly, this correspondence is as follows. *Adjacency sketches* and *probabilistic universal graphs* (PUGs) are probabilistic versions of adjacency labeling schemes and universal graphs, defined recently in [Har20] (Definitions 1.3 and 1.4). We map any communication problem f to a *hereditary* graph family $\mathfrak{F}(f)$, such that f has a constant-cost protocol if and only if $\mathfrak{F}(f)$ has a constant-size PUG (i.e. a constant-size adjacency sketch). If $\mathfrak{F}(f)$ has a constant-size PUG, then it is a positive example to the IGC (Proposition 1.6). On the other hand, we map any hereditary family $\mathcal{F}$ to a communication problem $\textsc{Adj}_{\mathcal{F}}$ that captures the complexity of computing adjacency in graphs from $\mathcal{F}$, where $\textsc{Adj}_{\mathcal{F}}$ has a constant-cost protocol if and only if $\mathcal{F}$ has a constant-size PUG. We emphasize that constant-cost communication problems always correspond to positive examples to the IGC, so determining the conditions for constant-cost communication becomes an equivalent question about the IGC:

Question 1.1. *What are the necessary and sufficient conditions for a positive example to the IGC to admit a constant-size PUG (i.e. a constant-size adjacency sketch)?*

This paper: We initiate the study of this question. From earlier work on PUGs [Har20] (which did not focus on the constant-size regime), we know that the set of families with constant-size PUGs is non-trivial: e.g. it includes the planar graphs but excludes the interval graphs. Our contributions are as follows. See Section 1.3 for a more detailed discussion.

1. We study an extensive list of families $\mathcal{F}$ that are important for the IGC and resolve Question 1.1 for the set of hereditary subfamilies $\mathcal{H} \subseteq \mathcal{F}$. We consider monogenic bipartite families, interval

[1] In this paper, randomized communication is always in the *public-coin, two-way* model; see Definition 2.1.

[2] $\textsc{Equality}(x, y) = 1$ if and only if $x = y$. GREATER-THAN takes value 1 on inputs $i, j \in [n]$ iff $i \le j$.

graphs, permutation graphs, families of bounded twin-width (including bounded tree- and clique-width), and many others (see Section 1.3). In each case, we find that $\mathcal{H} \subseteq \mathcal{F}$ admits a constant-size PUG if and only if it is *stable*. Stability (e.g. [CS18, NMP+21, GPT21]) is a property related to *monadic stability* in model theory. Informally, a family is *stable* if it does not encode the GREATER-THAN problem; for example, planar graphs are stable while interval graphs are not stable. Since GREATER-THAN has non-constant cost, stability is a necessary condition in Question 1.1. We are led to the conjecture that it is also sufficient (Conjecture 1.2).

2. We use constant-size PUGs to find new positive examples to the IGC. In particular, we show that constant-size PUGs (i.e. adjacency sketches) are preserved by the Cartesian product: if $\mathcal{F}$ has a constant-size PUG then so do the induced subgraphs of products of graphs in $\mathcal{F}$ (more generally, constant-size small-distance sketches are preserved). Therefore the induced subgraphs of any Cartesian product G^d are positive examples to the IGC. Despite the central importance of graph products in the study of labeling schemes (e.g. [CLR20]), it was not known whether these families even have factorial speed. This shows that PUGs are a powerful technique for studying the IGC. In fact, we show that this result *cannot* be obtained via some standard techniques (discussed below). We also establish that *stable* P_7-free bipartite graphs are positive examples to the IGC; the general case is an open problem [LZ17].
3. As a consequence of our results, we obtain constant-size *small-distance sketches* for planar graphs (and more generally *stable, bounded twin-width* graphs [GPT21]), answering an open question of [Har20]. Here we mean that for any constant k and any planar graph G, there is a constant-size random sketch sk such that $\mathsf{dist}(x, y) \leq k$ can be decided from $\mathsf{sk}(x)$, $\mathsf{sk}(y)$ with high probability (without knowing G). This implies an $O(\log n)$ *small-distance labeling scheme* (as in e.g. [ABR05]). More generally, for any *first-order formula* $\phi(x, y)$, we obtain a constant-size sketch for $\phi(x, y)$ and a $O(\log n)$ labeling scheme for deciding ϕ.
4. We observe that some standard techniques for approaching the IGC are equivalent to communication protocols with access to certain *oracles*. One standard technique (e.g. [KNR92, Cha18, CLR20]), which we call *equality-based labeling*, is equivalent to constant-cost deterministic communication with an EQUALITY oracle, which we call *equality-based communication*. Recent work [CLV19, BBM+20, PSW20, HHH21] has compared equality-based communication to randomized communication. We show that equality-based communication cannot compute adjacency in the hypercube. Therefore, it cannot compute k-HAMMING DISTANCE ([HSZZ06, BBG14, Sağ18]), and equality-based labeling cannot recover our results for Cartesian products. This correspondence also gives a very simple proof for the non-trivial fact that equality-based labeling cannot succeed for any family that is not stable.

Conclusions: While the interpretation of a communication problem (i.e. Boolean matrix) as a graph is standard (e.g. [HHH21]), the further step of constructing a *hereditary graph family* is especially useful for studying *constant-cost* communication, since a constant-cost protocol is inherited by induced subgraphs (submatrices). For every example studied in this paper, we have found that the hereditary families corresponding to constant-cost problems are exactly those that are *stable*. We conjecture that this always holds. We think of our conjecture as the probabilistic version of the IGC, and emphasize the similarity (see Figure 2):

Conjecture 1.2 (Probabilistic Universal Graph Conjecture)**.** *Any hereditary graph family $\mathcal{F}$ admits a constant-size PUG (i.e. a constant-size adjacency sketch) if and only if it is* stable *and has at most factorial speed.*

This conjecture and the IGC do not directly imply the other, but Conjecture 1.2 would establish the IGC for all *stable* families (Proposition 1.6). The IGC is challenging because there are no

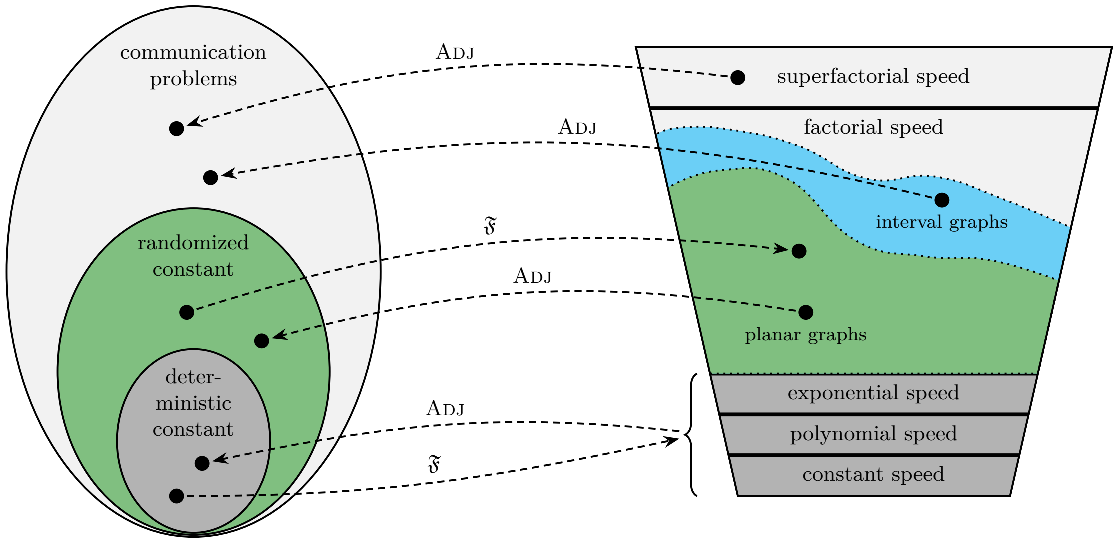

Figure 1: The correspondence that motivates this paper (Proposition 1.6). Section 3 describes the lattice on the right. Communication problems with constant-cost randomized protocols are mapped to the set of hereditary graph families with constant-size PUGs (and therefore poly(n) universal graphs by Proposition 1.9) by $\mathfrak{F}$. Families with constant-size PUGs are mapped to constant-cost communication problems by Adj.

known non-trivial structural results that hold for the entire set of factorial graph families [LZ15], and Conjecture 1.2 appears challenging for the same reason. But Conjecture 1.2 is a natural step towards the IGC because (a) imposing the stability condition seems to drastically simplify the structure of a hereditary family; (b) it gives a structural characterization of constant-cost communication problems; and (c) it captures the power of the probabilistic method for making progress on the IGC.

Conjecture 1.2 has unintuitive consequences for communication complexity: it implies that computing adjacency in graphs from a hereditary, factorial family can have complexity $O(1)$ or $\Omega(\log\log n)$ but *nothing in between* (Proposition 2.2); see Example 1.13. This is due to the *hereditary* property of $\mathcal{F}$; hereditary graph families exhibit similar gaps in their speed (see Section 3).

Outline. Section 1.1 states the main concepts and the correspondence between communication problems and hereditary graph families that motivates this paper. Section 1.2 gives some examples. Section 1.3 states our main results. Section 2 states basic results on PUGs. Section 3 reviews structural graph theory. The remaining sections prove our results.

1.1 Communication-to-Graph Correspondence

We now describe the central communication-to-graph correspondence that motivates our work, which is illustrated in Figure 1 and Figure 2. For any undirected graph $G = (V, E)$, we identify E with the function $E : V \times V \to \{0, 1\}$ where $E(x, y)$ holds true ($E(x, y) = 1$) if and only if (x, y) is an edge of G. In this paper, bipartite graphs $G = (X, Y, E)$ are colored; i.e. they are defined with a fixed partition of the vertices into parts X and Y.

A *communication problem* is a sequence $f = (f_n)_{n\in\mathbb{N}}$ of functions[3] $f_n : [n] \times [n] \to \{0, 1\}$. For any communication problem $f = (f_n)_{n\in\mathbb{N}}$, write $\mathsf{CC}(f_n)$ for the cost of the optimal two-way,

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

randomized protocol (Definition 2.1) computing f_n, and write $\mathsf{CC}(f)$ for the function $n \mapsto \mathsf{CC}(f_n)$. We may represent f_n as a bipartite graph $F_n = ([n], [n], f_n)$ where f_n defines the edge relation.

We will define the hereditary family $\mathfrak{F}(f)$ associated with f as the smallest hereditary family that contains each F_n, as follows. For graphs G, H, we write $H \sqsubset G$ if H is an induced subgraph of G. For any set of graphs $\mathcal{G}$, we define the *hereditary closure* $\mathsf{cl}(\mathcal{G}) := \{H : \exists G \in \mathcal{G}, H \sqsubset G\}$. By definition, $\mathsf{cl}(\mathcal{G})$ is a hereditary graph family that includes $\mathcal{G}$. We then define

$$\mathfrak{F}(f) := \mathsf{cl}(\{F_1, F_2, \dots\}) .$$

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 $G \in \mathcal{F}$. A communication problem contains only one function f_n for each input size n, whereas $\mathcal{F}$ contains many graphs of size n. The ADJACENCY problem should capture the maximum complexity of computing adjacency in any graph $G \in \mathcal{F}$, so for each input size $n \in \mathbb{N}$, we choose from $\mathcal{F}_n$ the graph where adjacency is hardest to compute. Let $\prec$ be a total order on functions $[n] \times [n] \to \{0,1\}$ that extends the partial order $\prec'$ defined by $f_n \prec' g_n \iff \mathsf{CC}(f_n) < \mathsf{CC}(g_n)$. We define the ADJACENCY problem as $\textsc{Adj}_{\mathcal{F}} = (f_n)_{n \in \mathbb{N}}$, where

$$f_n = \max\{g_n : ([n], g_n) \in \mathcal{F}_n\}$$

and the maximum is with respect to $\prec$. Here we have written $g_n : [n] \times [n] \to \{0,1\}$ for the edge relation in the graph $([n], g_n)$. It follows that for any communication problem f, we have $f = \textsc{Adj}_{\mathcal{F}}$ where $\mathcal{F} = \{F_1, F_2, \dots\}$, since $\mathcal{F}_n$ is a singleton; but it is *not* true that $f = \textsc{Adj}_{\mathfrak{F}(f)}$, since for each $n \in \mathbb{N}$, $\textsc{Adj}_{\mathfrak{F}(f)}$ effectively chooses the *hardest* subproblem of size n of any f_m for $m \geq n$. In this way, the map $\mathfrak{F}(f)$ has the effect of "blowing up" non-constant subproblems (see Example 1.13), which makes it effective for studying the constant-cost problems where this has no effect.

To proceed with the correspondence, we need *probabilistic universal graphs* and *stability*.

1.1.1 Probabilistic Universal Graphs & Adjacency Sketches

Universal graphs were introduced in [Rad64] and *adjacency labeling schemes* were introduced in [Mul89, KNR92]. As observed in [KNR92], these concepts are equivalent. By analogy, [Har20] defined PUGs[4] and the equivalent *randomized* adjacency labeling schemes, which we call *adjacency sketches* in this paper. 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 family to a PUG that supports adjacency queries.

Definition 1.3 (Probabilistic Universal Graph, [Har20])**.** A *probabilistic universal graph sequence* for a graph family $\mathcal{F}$ (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 $\phi : V(G) \to V(U_n)$ such that

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

Definition 1.4 (Adjacency Sketch)**.** For a graph family $\mathcal{F}$, 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 $\mathsf{sk} : V(G) \to \{0,1\}^{c(n)}$. The encoder and (deterministic) decoder $D : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ satisfy the condition that for all $G \in \mathcal{F}$,

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

[4] A similar idea is mentioned briefly in [FK09].

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 $\mathsf{SK}(\mathcal{F})$ for the smallest function $c(n)$ such that there is an adjacency sketch for $\mathcal{F}$ with cost $c(n)$ and error $\delta = 1/3$.

It was observed in [Har20] that PUGs are equivalent to adjacency sketches, by the same argument as in [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 $\mathsf{sk} : V(G) \to \{0,1\}^{c(n)}$ with the map $\phi : V(G) \to V(U_n)$, and identify the decoder D with the edge relation on U_n. We get:

Proposition 1.5. *A hereditary family $\mathcal{F}$ has a constant-size PUG if and only if $\mathsf{SK}(\mathcal{F}) = O(1)$.*

These objects were introduced in [Har20] to model communication problems where two parties send single messages to a referee, who computes $f(x, y)$ from the messages without knowing f in advance; no special attention was paid to the constant-cost regime that we study in this paper. We use these objects to establish the following properties of $\mathfrak{F}$ and Adj, thereby associating constant-cost communication problems with hereditary graph families that admit constant-size PUGs. The (simple) proof of this statement is in Section 2.2.

Proposition 1.6. *For any communication problem $f = (f_n)_{n\in\mathbb{N}}$ and hereditary graph family $\mathcal{F}$:*

1. *$\mathsf{CC}(f) = O(1)$ if and only if $\mathfrak{F}(f)$ has a constant-size PUG (i.e. $\mathsf{SK}(\mathfrak{F}(f)) = O(1)$).*
2. *$\mathcal{F}$ has a constant-size PUG if and only if $\mathsf{CC}(\textsc{Adj}_{\mathcal{F}}) = O(1)$.*

We now connect constant-size PUGs to the IGC. A quantitative version of equivalence between universal graphs and labeling schemes is as follows (and the IGC is its converse):

Proposition 1.7 ([KNR92])**.** *If a hereditary graph family $\mathcal{F}$ has a universal graph of size $2^{O(\log n)} = \mathrm{poly}(n)$ (i.e. an adjacency labeling scheme of size $O(\log n)$), then $|\mathcal{F}_n| \leq \binom{\mathrm{poly}(n)}{n} = 2^{O(n \log n)}$.*

Using a derandomization result of [Har20], we see that constant-size adjacency sketches imply adjacency labeling schemes of size $O(\log n)$ (we give a much simpler proof in Section 2).

Lemma 1.8 ([Har20])**.** *For any hereditary graph family $\mathcal{F}$, there is an adjacency labeling scheme of size $O(\mathsf{SK}(\mathcal{F}) \cdot \log n)$.*

It follows that hereditary graph families which admit constant-size PUGs are positive examples to the IGC, as illustrated in Figure 1. We refine this inclusion in Section 1.1.2.

Proposition 1.9. *If a hereditary family $\mathcal{F}$ has a constant-size PUG (i.e. $\mathsf{SK}(\mathcal{F}) = O(1)$) then it is a positive example to the IGC (i.e. it admits a universal graph of size $\mathrm{poly}(n)$).*

1.1.2 Chain Number & Stability

To refine the correspondence, we introduce the notions of *chain number* and *stability*.

Definition 1.10 (Chain Number)**.** For a graph G, the *chain number* $\mathsf{ch}(G)$ 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 family $\mathcal{F}$, we write $\mathsf{ch}(\mathcal{F}) = \max_{G\in\mathcal{F}} \mathsf{ch}(G)$. If $\mathsf{ch}(\mathcal{F}) = \infty$, then $\mathcal{F}$ has *unbounded chain number*, otherwise it has *bounded chain number*.

As in [CS18, NMP+21] we call graph families of bounded chain number *stable* (they are also called *graph-theoretically stable* in [GPT21]). These families have many interesting properties, including stronger versions of Szemerédi's Regularity Lemma [MS14] and the Erdős-Hajnal property [CS18] (it is also conjectured in [HHH21] that graphs which admit constant-cost protocols for adjacency satisfy the *strong* Erdős-Hajnal property), and they play a central role in algorithmic

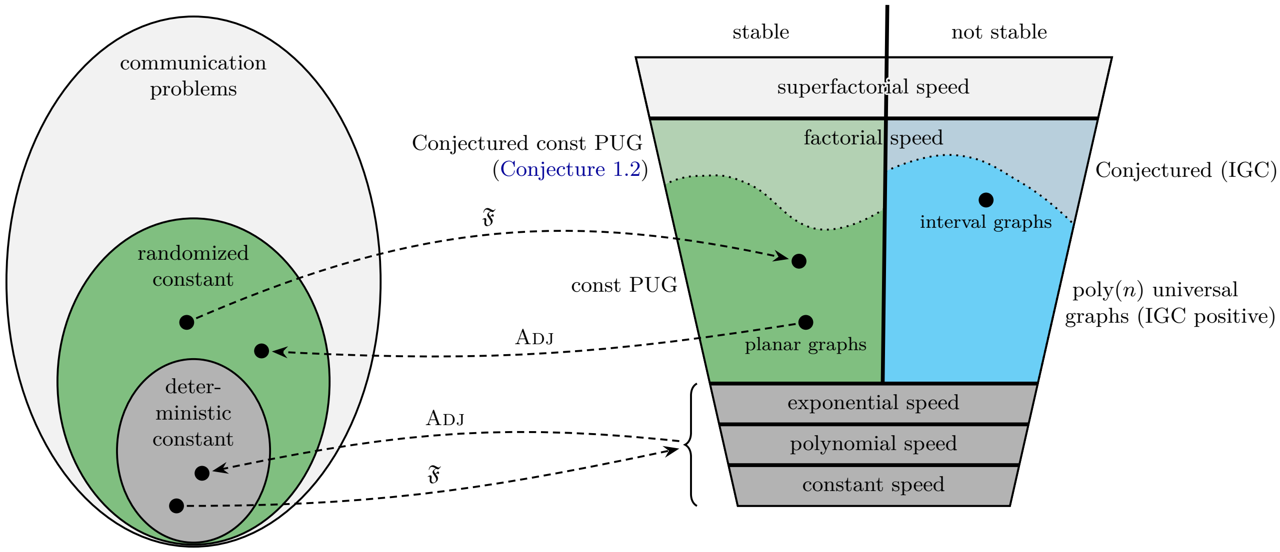

Figure 2: The stability condition partitions the factorial layer. Stable families fall under Conjecture 1.2, while the remaining factorial families fall under the IGC.

graph theory [GPT21]. The present paper shows that stability is also essential for understanding the IGC and randomized communication complexity.

This is clearly illustrated by the GREATER-THAN problem, which is defined as $\mathrm{GT}_n : [n] \times [n] \to \{0, 1\}$, where $\mathrm{GT}_n(x, y) = 1$ if and only if $x \le y$. It is straightforward to check that, if a hereditary family $\mathcal{F}$ has unbounded chain number, then the GREATER-THAN problem on domain $[n]$ can be reduced to the problem of computing adjacency in some graph $G \in \mathcal{F}_{2n}$. Using known bounds on the communication complexity of GREATER-THAN (in the *simultaneous message passing* model; see Section 2), we may then conclude:

Proposition 1.11. *If a hereditary graph family $\mathcal{F}$ is not stable, then* $\mathsf{SK}(\mathcal{F}) = \Omega(\log n)$.

This is proved formally in Section 2.2. Therefore, for a hereditary graph family $\mathcal{F}$ to have a constant-size PUG, it is necessary that it has bounded chain number, i.e. it is stable. In other words, if f is any communication problem with $\mathsf{CC}(f) = O(1)$ then $\mathfrak{F}(f)$ is stable.

We can now state a refined communication-to-graph correspondence, illustrated in Figure 2. It states that the constant-cost communication problems are equivalent to the hereditary graph families $\mathcal{F}$ that admit a constant-size PUG, which is a set of positive examples to the IGC that are stable (Conjecture 1.2 states that it is *exactly* this set).

Proposition 1.12. *If $\mathcal{F}$ has a constant-size PUG (i.e. $\mathsf{SK}(\mathcal{F}) = O(1)$) then $\mathcal{F}$ is stable, and it is a positive example to the IGC (i.e. $|\mathcal{F}_n| = 2^{O(n \log n)}$ and it admits a universal graph of size* poly(n)*).*

We conclude this section with a useful characterization of stable graph families via forbidden induced subgraphs. It is a well-known fact that any hereditary graph family can be defined by its set of *minimal forbidden induced subgraphs*. That is, for any hereditary family $\mathcal{F}$, there is a *unique minimal* set of graphs $\mathcal{H}$ such that $\mathcal{F}$ is the family *$\mathcal{H}$-free* graphs, i.e. $\mathcal{F} = \mathsf{Free}(\mathcal{H})$, where

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

One can show (Proposition 3.5) that a graph family $\mathcal{F}$ has a bounded chain number (i.e. $\mathcal{F}$ is stable) if and only if

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

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* (Definition 3.2, depicted in Figure 3). For any $\mathcal{F}$, we denote by $\mathsf{stable}(\mathcal{F})$ the set of all stable subfamilies of $\mathcal{F}$.

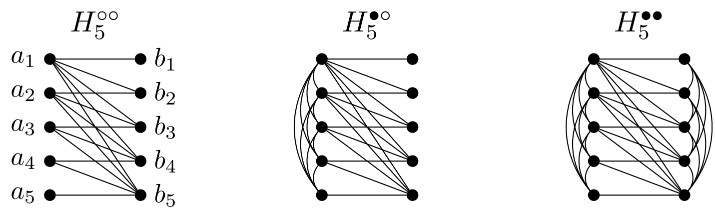

Figure 3: Examples of the half-graph, co-half-graph, and threshold graphs.

1.2 Examples

Before stating our results, we give two examples to clarify this correspondence and Conjecture 1.2.

Example 1.13. The k-Hamming Distance problem HD^k requires Alice and Bob to decide whether the Hamming distance between their inputs $x, y \in \{0,1\}^d$ is at most $k(d)$. It has complexity $\Theta(k(d)\log k(d))$ when $k(d) = o(\sqrt{d})$ [HSZZ06, Sağ18]. Setting $k(d)$ to be non-constant, Conjecture 1.2 demands that the hereditary graph family obtained from $\mathfrak{F}$ either has unbounded chain number, or has superfactorial speed. To show that it has unbounded chain number, we show that for every $t \in \mathbb{N}$ we can choose d sufficiently large to construct disjoint sets $\{a^{(1)}, \dots, a^{(t)}\}, \{b^{(1)}, \dots, b^{(t)}\} \subset \{0,1\}^d$ so that $\mathsf{dist}(a_i, b_j) \le k(d)$ if and only if $i \le j$. We choose d such that $k = k(d) \ge t$ and define $a_r^{(i)} = 1$ if and only if $r = i$ and $b_r^{(j)} = 1$ if and only if $r > d - k + j$ or $r \le j$.

For $i \le j$ it holds that $a_i^{(i)} = b_i^{(j)} = 1$ while $b^{(j)}$ takes value 1 on exactly $k-1$ other coordiantes, so $\mathsf{dist}(a^{(i)}, b^{(j)}) \le k$. On the other hand, if $i > j$ then $a_i^{(i)} = 1, b_i^{(j)} = 0$ and $b^{(j)}$ takes value 1 on exactly k other coordinates, so $\mathsf{dist}(a^{(i)}, b^{(j)}) = k + 1$.

This illustrates some subtleties of the correspondence. Write $n = 2^d$ and think of domain $\{0,1\}^d$ as $[n] = [2^d]$. Let $k(d) = \omega(1)$. Then $\mathsf{CC}(\text{HD}_n^k) = \Theta(k(d)\log k(d))$. The corresponding hereditary graph family $\mathfrak{F}(\text{HD}^k)$ has unbounded chain number, so for every m there is $G \in \mathfrak{F}(\text{HD}^k)$ with chain number m. So $\mathsf{CC}(\textsc{Adj}_{\mathfrak{F}(\text{HD}^k)}) = \Omega(\log d) = \Omega(\log\log n)$. But for $k(d) = \log\log\log d$, say, this is a doubly-exponential increase in complexity. This shows how the hereditary closure within the map $\mathfrak{F}$ "blows up" any non-constant subproblem.

Example 1.14. A projective plane is a set of points P and a set of lines L (which are subsets of points), where any two points lie on a unique line, and any two lines intersect at a unique point. Consider the communication problem where Alice receives a point $p \in P$ and Bob receives a line $\ell \in L$, and they must decide whether $p \in \ell$. The complexity depends on the choice of P and L. One may interpret this problem as computing adjacency in a *projective plane graph*; these graphs form a hereditary family $\mathcal{P}$ (the C_4-free bipartite graphs). Then the maximum complexity of an instance of size $n = |P| + |L|$ is $\mathsf{CC}((\textsc{Adj}_{\mathcal{P}})_n)$. $\mathcal{P}$ is stable, so we do not get a non-constant lower bound from the Greater-Than problem. But it is known that $\mathcal{P}$ has superfactorial speed (see e.g. [LZ15]), which rules out a constant-cost protocol.

1.3 Results

Equipped with the correspondence in Proposition 1.6 and Proposition 1.12, we are now prepared to state our results towards an answer of Question 1.1. Our goal is to characterize the set of hereditary graph families (necessarily with factorial speed) that admit constant-size adjacency sketches

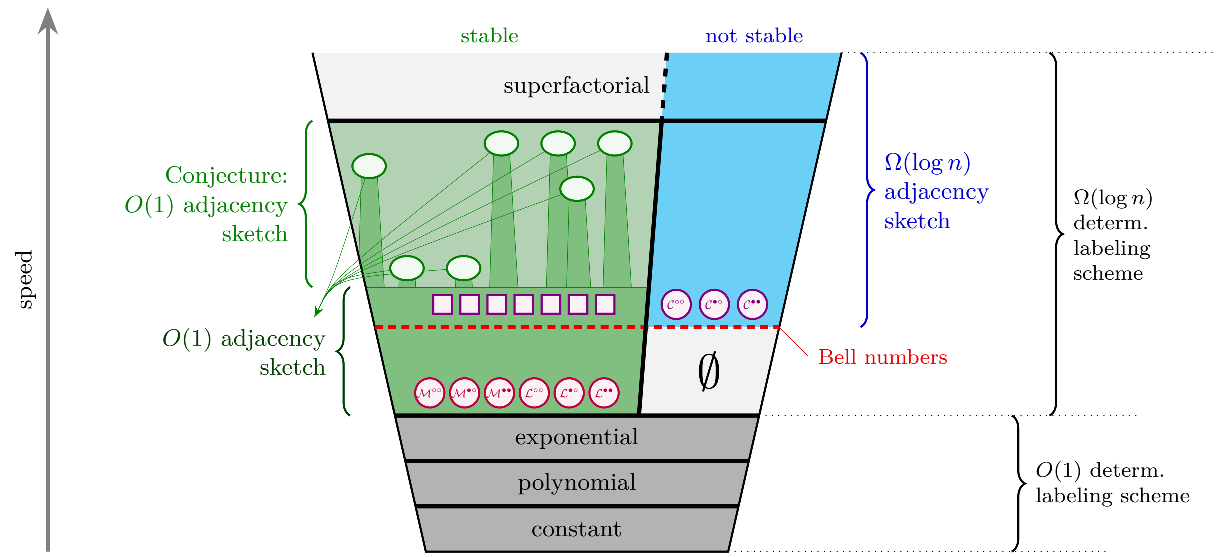

Figure 4: Overview of our results (in green). Section 3 describes the lattice on the right. Circles are minimal factorial families; purple shapes are minimal families above the Bell numbers.

(i.e. constant-size PUGs); these families correspond exactly to the constant-cost communication problems. Most of our results are of the following form: we choose a hereditary family $\mathcal{F}$ with factorial speed, and characterize the subfamilies of $\mathcal{F}$ that admit constant-size PUGs. In each case, we find that these subfamilies are exactly the set $\mathsf{stable}(\mathcal{F})$. See Figure 4 for an illustration.

Section 3: Contextual Results. Our first set of results place Question 1.1 in context of structural graph theory and motivate the formulation of Conjecture 1.2. These results are the basis of Figure 4; they are relatively simple, relying mostly on known structural results. In particular, 6 out of 9 *minimal factorial families* are stable, and they admit constant-size PUGs. Likewise, all but 3 *minimal families above the Bell numbers* are stable, and they each admit a constant-size PUG, along with all hereditary graph families that fall below the Bell numbers.

Section 4: Bipartite Graphs. The map $\mathfrak{F}$ transforms a communication problem into a hereditary family of *bipartite* graphs. In Lemma 4.1, we show that proving Conjecture 1.2 for families of bipartite graphs suffices to prove the full conjecture. Since any hereditary subfamily of bipartite graphs can be defined by forbidding a set $\mathcal{H}$ of bipartite graphs, the natural first step towards proving Conjecture 1.2 for bipartite graph families is to prove it for the *monogenic* bipartite graph families where $\mathcal{H}$ is a singleton.

Theorem 1.15. *Let H be a bipartite graph such that the family of H-free bipartite graphs is factorial. Then any hereditary subfamily $\mathcal{F}$ 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 families of bipartite graphs. Previous work [All09, LZ17] has shown that a family 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 4). 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 of [LZ17] for P_7-free graphs.

As a result, we get a poly(n)-size universal graph for any stable subfamily of the P_7-free bipartite graphs. The P_7-free bipartite graphs are a factorial family, but constructing a poly(n)-size universal graph (as predicted by the IGC) is an open problem [LZ17]. We take this as evidence that, independent of randomized communication, the stable graph families are an interesting special case of the IGC that might allow progress.

We remark that, under a conjecture of [LZ17], if our theorem was proved for the families of bipartite graphs obtained by excluding only *two* graphs H_1, H_2, it would establish Conjecture 1.2 for any family of bipartite graphs obtained by excluding any *finite* set of induced subgraphs.

Section 5: Interval & Permutation Graphs. Two typical examples of graph families where the IGC holds are the interval graphs and permutation graphs. Interval graphs were used in [Har20] as a non-trivial example where adjacency sketches must be asymptotically as large as the adjacency labels. In our terminology, this is because interval graphs are not stable. Permutation graphs are also not stable, so do not admit constant-size adjacency sketches. Our goal is to determine how much these graphs must be restricted before they admit constant-size adjacency sketches (i.e. PUGs).

Theorem 1.16. *Let $\mathcal{F}$ be any hereditary subfamily 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 graphs[5]. We show that any stable subfamily of interval graphs can be reduced to a family with bounded treewidth, and therefore has a constant-size PUG (implied by Lemma 2.13). A consequence of our proof is that stable subfamilies 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.

Section 6: Cartesian Products. Denote by $G \square H$ the Cartesian product of G and H, and write $G^{\square d}$ for the d-wise product of G. For example, $P_2^{\square d}$ is the 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 are positive examples to the IGC. We resolve this question by proving that Cartesian products preserve constant-size PUGs and distance-k sketches (which compute $\mathsf{dist}(x, y) \le k$). Write $\mathcal{F}^{\square} := \{G_1 \square \cdots \square G_d : d \in \mathbb{N}, G_i \in \mathcal{F}\}$.

Theorem 1.17. *If $\mathcal{F}$ is any family that admits a constant-size PUG (including any finite family), then $\mathsf{cl}(\mathcal{F}^{\square})$ admits a constant-size PUG. For any fixed k, if $\mathcal{F}$ admits a constant-size distance-k sketch, then so does $\mathcal{F}^{\square}$.*

This implies that $\mathsf{cl}(\mathcal{F}^{\square})$ is a positive example to the IGC whenever $\mathcal{F}$ admits a constant-size PUG (e.g. when $\mathcal{F}$ is finite), and that for any constant k there is a deterministic $O(\log n)$ labeling scheme for deciding $\mathsf{dist}(x, y) \le k$ in $\mathcal{F}^{\square}$, whenever $\mathcal{F}$ admits a constant-size distance-k sketch. Other common graph products – the strong, direct, and lexicographic products (see e.g. [HIK11]) – either admit trivial constant-size PUGs or have unbounded chain number.

[5] A 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 [GPT21], which we use for our twin-width result.

Section 7: Equality-Based Communication & Labeling. The EQUALITY problem is the standard example of a constant-cost communication problem [NK96, RY20]. Recent work [CLV19] asked whether efficient (i.e. poly($\log\log n$) on domain $[n]$) randomized communication can always be simulated by an efficient deterministic protocol with access to an EQUALITY oracle; we call such protocols *equality-based* (see also [PSW20, BBM+20, HHH21]). [CLV19] gave a problem that admits an $O(\log\log n)$ randomized protocol, but any equality-based protocol has cost $\Omega(\log n)$. This leaves open the natural question of whether *constant-cost* protocols can be simulated by the EQUALITY oracle. We show[6] there is no constant-cost equality-based protocol for adjacency in the hypercube P_2^n; i.e. there is no equality-based protocol for 1-HAMMING DISTANCE, so the EQUALITY oracle cannot simulate the k-HAMMING DISTANCE protocol [HSZZ06, BBG14, Sağ18].

Theorem 1.18. *There is no constant-cost equality-based protocol for computing adjacency in P_2^n.*

An interesting connection is that equality-based communication corresponds to a standard type of adjacency labeling scheme (e.g. [KNR92, CLR20]; [Cha18] defines a weaker type), which we call *equality-based labeling* (Definition 2.5). All labeling schemes given in this paper are of this type, except (by necessity) for Cartesian products. Informally, labels are constructed as a question-and-answer series: for each question, a partition $\mathcal{P}$ of vertices is chosen in advance. The label for x asks "Does y belong to $P \in \mathcal{P}$?", and the label for y gives the answer. An example is the simple scheme for the family of forests: $\mathcal{P}$ is the partition into singletons, and the label for x asks "Is y the parent of x?" and the label for y asks "Is x the parent of y"? Our correspondence also gives a simple proof (Proposition 2.9) that equality-based labeling fails for any family that is not stable.

Section 8: Twin-Width & Distance Sketching. Graph width parameters like treewidth [RS86] and clique-width [CER93] are a central tool in structural graph theory. Often, graph families where a certain width parameter is bounded possess favorable structural, algorithmic, or combinatorial properties. The powerful *twin-width* parameter, introduced recently in [BKTW20], generalizes treewidth and clique-width and has attracted a lot of recent attention [GPT21, SS21, AHKO21, BH21, ST21]. Graph families of bounded twin-width are positive examples to the IGC [BGK+21], making them a natural choice for studying Question 1.1. Building upon recent structural results on stable families of bounded twin-width [GPT21], we prove:

Theorem 1.19. *Let $\mathcal{F}$ be a hereditary family of graphs with 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 sketch that generalizes adjacency and distance-k sketches. For any *first-order formula* $\phi(x, y)$ on vertex pairs (see Section 8), we consider sketches that allow to decide $\phi(x, y)$ instead of adjacency. An example is the $\mathsf{dist}(x, y) \le k$ formula:

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

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 1.19:

Corollary 1.20. *Let $\mathcal{F}$ be a stable family 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 gives us constant-size distance-k sketches for stable families of bounded twin-width. This answers an open question of [Har20], who asked about distance-k sketches for planar graphs (which are stable, because they have bounded degeneracy, and are of bounded twin-width [BKTW20]).

[6] After preparation of this manuscript, we found that this result was also proved independently with a very different Fourier-analytic technique in the recent preprint [HHH21].

2 Preliminaries

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.

All graphs in this work are simple, i.e. undirected, without loops and multiple edges. Let G be a graph and let v be a vertex in G. A vertex that is adjacent to v is called a *neighbour* of v. The set of all neighbours of v is called the *neighbourhood* of v and it is denoted as $N(v)$. The *degree* of v is the number of neighbours of v and it is denoted as $\deg(v)$. A bipartite graph is a graph whose vertex set can be partitioned into two independent sets. 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 $E : X \times Y \to \{0, 1\}$ 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.

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 $\phi : X' \cup Y' \to X \cup Y$ 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 any graph $G = (V, E)$ and subset $W \subseteq V$, we write $G[W]$ for the subgraph of G induced by W. For 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 $(x, y) \in E$.

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

The *disjoint union* of two graphs $G = (V, E)$ and $H = (V', E')$ is the graph $G + H = (V \cup V', E \cup E')$.

2.2 Communication Complexity

We refer the reader to [NK96, RY20] for an introduction to communication complexity. For a function $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, we will write $\mathsf{CC}(f)$ for the optimal two-way, public-coin randomized communication cost of f. Informally, in this model, the players Alice and Bob share a source of randomness. Alice receives input x, Bob receives input y, 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 $b = f_n(x, y)$ with probability at least $2/3$. The cost of such a protocol is the maximum over all inputs 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 over *communication trees*. For input size n, a communication tree T_n is a binary tree with each inner node being a tuple (p, m) where $p \in \{A, B\}$ and $m : [n] \to \{0, 1\}$, and each edge of T_n is labeled either 0 or 1. Each leaf node is labeled either 0 or 1. For any fixed tree T_n and inputs $x, y \in [n]$, 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, m) then Alice sends $m(x)$ to Bob and both players set c to the child along the edge labeled $m(x)$. If c is an inner node (B, m) then Bob sends $m(y)$ to Alice, and both players set c to the child along the edge labeled $m(y)$. 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.

For communication problem $f = (f_n)_{n\in\mathbb{N}}$, a randomized protocol must satisfy $T_n(x, y) = f_n(x, y)$ with probability at least $2/3$, where the probability is over the choice of T_n. The cost of a protocol for f_n is the minimum value d such that all trees T_n in the support of the distribution have depth at most d.

Proposition 1.6. *For any communication problem $f = (f_n)_{n\in\mathbb{N}}$ and hereditary graph family $\mathcal{F}$:*

1. *$\mathsf{CC}(f) = O(1)$ if and only if $\mathfrak{F}(f)$ has a constant-size PUG (i.e. $\mathsf{SK}(\mathfrak{F}(f)) = O(1)$).*
2. *$\mathcal{F}$ has a constant-size PUG if and only if $\mathsf{CC}(\textsc{Adj}_{\mathcal{F}}) = O(1)$.*

Proof. First suppose that $\mathcal{F}$ is a hereditary graph family with $\mathsf{SK}(\mathcal{F}) = O(1)$, and write $\textsc{Adj} = (\textsc{Adj}_n)_{n\in\mathbb{N}}$ for the communication problem $\textsc{Adj}_{\mathcal{F}}$. 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 $\textsc{Adj}$ as follows. For each $n \in \mathbb{N}$, let $G_n \in \mathcal{F}_n$ be the graph such that $\textsc{Adj}_n$ is the edge relation of G_n. On inputs $x, y \in [n]$, Alice and Bob sample $\mathsf{sk} \sim \Phi_{G_n}$ and Alice sends $\mathsf{sk}(x)$ to Bob, which requires at most $\mathsf{SK}(\mathcal{F})$ bits of communication. Then Bob simulates the decoder on $D(\mathsf{sk}(x), \mathsf{sk}(y))$. By definition

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

Now suppose that $\mathsf{CC}(\textsc{Adj}) = O(1)$. For any $G \in \mathcal{F}_n$ it holds that the edge relation $g_n : [n] \times [n] \to \{0, 1\}$ for G satisfies $\mathsf{CC}(g_n) \leq \mathsf{CC}(\textsc{Adj}_n)$, by definition. 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 $\mathsf{CC}(\textsc{Adj})$. So there is some d, such that, for every $G \in \mathcal{F}$, all communication trees T in the support of $\mathcal{P}(G)$ have 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 $\mathsf{sk}(v)$ the following:
>
> 1. If c is an inner node (p, m) (with $p \in \{A, B\}$ and $m : [n] \to \{0, 1\}$), 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 $(\mathsf{sk}(u), \mathsf{sk}(v))$, the decoder simulates the communication tree T on (u, v) using the values $m(u), m(v)$ for each inner node. We therefore obtain

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

From the first argument above, it is clear that for any communication problem f, if $\mathsf{SK}(\mathfrak{F}(f)) = O(1)$ then $\mathsf{CC}(f) = O(1)$. In the other direction, assume that $\mathsf{CC}(f) \leq d$ for some constant d, and consider $\mathsf{SK}(\mathfrak{F}(f))$, where $\mathfrak{F}(f) = \mathsf{cl}(\{F_1, F_2, \dots\})$ for the graphs F_n on vertex set $[n]$ with edge relation f_n. Then it holds for any $G \in \mathfrak{F}(f)$ that there exists $n \in \mathbb{N}$ such that $G \sqsubset F_n$. But then the edge relation g of G satisfies $\mathsf{CC}(g) \leq \mathsf{CC}(f_n) \leq d$, since the communication problem g is a subproblem of f_n. We may then construct adjacency sketches by the scheme above, so we conclude $\mathsf{SK}(\mathfrak{F}(f)) = O(1)$. ☐

For the purpose of the next two statements, we briefly describe the public-coin *simultaneous message passing (SMP)* model of communication. In this model, given (private) inputs $x, y \in [n]$ to problem $f_n : [n] \times [n] \to \{0,1\}$, 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 $\Theta(\log n)$ (see the bibliographic remark in Appendix C).

Proposition 1.11. *If a hereditary graph family $\mathcal{F}$ is not stable, then $\mathsf{SK}(\mathcal{F}) = \Omega(\log n)$.*

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 $\mathrm{GT}_n(x,y)$ in the SMP model by choosing a graph $G \in \mathcal{F}$ with $\mathsf{ch}(G) = n$, 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 $\mathsf{sk}(a_x), \mathsf{sk}(b_y)$ according to the adjacency sketch for H, and send them to the referee, who outputs $D(\mathsf{sk}(a_x), \mathsf{sk}(b_y))$. This communication protocol has complexity at most $\mathsf{SK}(\mathcal{F})$, so by the lower bound on the SMP complexity of Greater-Than, we must have $\mathsf{SK}(\mathcal{F}) = \Omega(\log n)$. ☐

The next proposition shows an unintuitive consequence of Conjecture 1.2. One may wonder if it is possible to construct communication problems $\textsc{Adj}_{\mathcal{F}}$ of any given complexity by choosing an appropriate hereditary family $\mathcal{F}$. Under Conjecture 1.2, this is not possible. This is explained by hereditary property of $\mathcal{F}$. As in Example 1.13, any non-constant cost subproblem is "blown up" to full size. We write $\mathsf{SMP}(\textsc{Adj}_{\mathcal{F}})$ for the complexity of adjacency in graphs $G \in \mathcal{F}$, where we redefine $\textsc{Adj}_{\mathcal{F}}$ to choose the problem for each domain size n that is hardest in the SMP model (instead of the two-way model as it was defined earlier).

Proposition 2.2 (Complexity gaps)**.** *Assume Conjecture 1.2. Then for any hereditary, factorial graph family $\mathcal{F}$, the following hold:*

1. *Either $\mathsf{CC}(\textsc{Adj}_{\mathcal{F}}) = O(1)$ or $\mathsf{CC}(\textsc{Adj}_{\mathcal{F}}) = \Omega(\log \log n)$; and*
2. *Either $\mathsf{SMP}(\textsc{Adj}_{\mathcal{F}}) = O(1)$ or $\mathsf{SMP}(\textsc{Adj}_{\mathcal{F}}) = \Theta(\log n)$.*

Proof sketch. If $\mathcal{F}$ is stable, then by the conjecture we have $\mathsf{SK}(\mathcal{F}) = O(1)$. Using the reduction from communication to adjacency sketching in Proposition 1.6, it holds that $\mathsf{CC}(\textsc{Adj}_{\mathcal{F}}) \leq \mathsf{SK}(\mathcal{F}) = O(1)$; similarly, using the reduction in Proposition 1.11, it holds that $\mathsf{SMP}(\textsc{Adj}_{\mathcal{F}}) \leq \mathsf{SK}(\mathcal{F}) = O(1)$. If $\mathcal{F}$ is not stable, then for every $n \in \mathbb{N}$ there is a graph $G \in \mathcal{F}_{2n}$ with $\mathsf{ch}(G) = n$. By the same reduction as in Proposition 1.11, we can reduce Greater-Than on domain $[n]$ to adjacency in G. Then $\mathsf{SMP}(\textsc{Adj}_{\mathcal{F}}) = \Omega(\log n)$ by the lower bound mentioned above, while $\mathsf{CC}(\textsc{Adj}_{\mathcal{F}}) = \Omega(\log \log n)$ due to the lower bound[7] of [Vio15, RS15]. ☐

2.3 Boosting & Derandomization

Here we state some elementary facts about adjacency sketches and PUGs.

[7]Recall that we use n for the domain size, whereas standard notation in communication complexity is to use n for the *number of bits* in the input.

Proposition 2.3 (Probability boosting)**.** *Let $\mathcal{F}$ be a family of graphs. For any $\delta \in (0, 1/2)$, there is an adjacency sketch with error δ and size at most $O(\mathsf{SK}(\mathcal{F}_n) \cdot \log \frac{1}{\delta})$. Equivalently, if there is a PUG $U = (U_n)$ for $\mathcal{F}$ with size $|U_n|$, then there is a PUG $U' = (U'_n)$ for $\mathcal{F}$ with error δ and $|U'_n| \le |U_n|^{O(\log(1/\delta))}$.*

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

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

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

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

The following statement was proven in [Har20]. We include a much simpler proof here for the sake of completeness.

Lemma 2.4 (Labeling derandomization)**.** *For any graph family $\mathcal{F}$, there is an adjacency labeling scheme with size at most $O(\mathsf{SK}(\mathcal{F}_n) \cdot \log n)$.*

Proof. Let $G \in \mathcal{F}_n$. Using Proposition 2.3 for $\delta = 1/n^3$, we obtain an adjacency sketch with error probability δ and size $c(n) = O(\mathsf{SK}(\mathcal{F}_n) \cdot \log(1/\delta)) = O(\mathsf{SK}(\mathcal{F}_n) \cdot \log n)$. For a fixed sketch function $\mathsf{sk} : V(G) \to \{0,1\}^{c(n)}$, write Δ_{sk} for the number of pairs x, y such that the decoder outputs the incorrect value on input $\mathsf{sk}(x), \mathsf{sk}(y)$. Then, by the union bound, for a randomly chosen sketch function sk,

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

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

$$\mathbb{P}[\Delta_{\mathsf{sk}} > 0] = \mathbb{P}[\Delta_{\mathsf{sk}} \ge 1] = \mathbb{P}[\Delta_{\mathsf{sk}} \ge n\mu] \le \frac{\mu}{n\mu} = 1/n ,$$

so a random sk function is a (deterministic) adjacency labeling scheme for G with probability at least $1 - 1/n$. □

2.4 Equality-Based Labeling Schemes

We introduce the concept of *equality-based labeling schemes*. For simplicity of notation, we write

$$\textsc{Eq}(a, b) = \mathbb{1}[a = b] .$$

Definition 2.5. (Equality-based Labeling Scheme). Let $\mathcal{F}$ be a family 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 $D_{p_1,p_2} : \{0,1\}^{k \times k} \to \{0,1\}$ 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 $Q_{x,y}(i,j) = \textsc{Eq}(q_i(x), q_j(y))$. If $s = 0$ we simply write $D(Q_{x,y})$.

Remark 2.6. 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 and the label consists of a constant number of tuples, it is straightforward to put such a label into the form required by Definition 2.5.

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.

Lemma 2.7. *Let $\mathcal{F}$ be any hereditary family that admits a constant-size equality-based labeling scheme. Then $\mathcal{F}$ admits a constant-size adjacency sketch (and hence a constant-size PUG).*

Proof. Suppose $\mathcal{F}$ admits an (s, k)-equality based labeling scheme. We construct a constant-size adjacency sketch for $\mathcal{F}$ as follows. For any graph $G \in \mathcal{F}_n$, observe that we may assume each equality code $q_i(x)$ is in $[n]$, since there are at most n vertices.

1. For each $t \in [n]$ we assign a uniformly random number $r(t) \sim [3k^2]$.

2. For each vertex x assign the label $\mathsf{sk}(x) = (p(x), r(q_1(x)), \dots, r(q_k(x)))$.

On input $\mathsf{sk}(x), \mathsf{sk}(y)$, the decoder constructs the matrix $R \in \{0,1\}^{k \times k}$ where

$$R(i,j) = \textsc{Eq}(r(q_i(x)), r(q_j(y))) ,$$

and outputs

$$D(\mathsf{sk}(x), \mathsf{sk}(y)) = D_{p(x),p(y)}(R) .$$

It suffices to show that, for any x, y, we will have $R = Q_{x,y}$ with probability at least $2/3$. By the union bound,

$$\mathbb{P}[R \neq Q_{x,y}] \leq \sum_{i,j \in [k]} \mathbb{P}[\textsc{Eq}(r(q_i(x)), r(q_j(x))) \neq \textsc{Eq}(q_i(x), q_j(y))] .$$

For any i, j, if $q_i(x) = q_j(y)$ we will have $r(q_i(x)) = r(q_j(y))$ with probability 1. On the other hand, if $q_i(x) \neq q_j(y)$ we have $r(q_i(x)) = r(q_j(y))$ with probability at most $1/3k^2$. Therefore

$$\mathbb{P}[R \neq Q_{x,y}] \leq k^2 \cdot \frac{1}{3k^2} = 1/3 . \qquad \square$$

We note that, although the construction of R in the above sketch has only one-sided error (i.e. if $\textsc{Eq}(q_i(x), q_j(y)) = 1$ then $R(i,j) = 1$ with probability 1), the adjacency sketch does *not* necessarily have one-sided error. This is because the matrix R is the input to an arbitrary function.

Equality-based labeling schemes imply constant-size sketches and PUGs by the above lemma, and by Lemma 2.4 this implies $O(\log n)$ adjacency labeling schemes and $\mathrm{poly}(n)$ universal graphs. However, in this particular case, the probabilistic method in Lemma 2.4 is not necessary, since each equality code can be written with $O(\log n)$ bits.

Proposition 2.8 (Naïve Derandomization)**.** *Let $\mathcal{F}$ be any hereditary family that admits a constant-size equality-based labeling scheme. Then $\mathcal{F}$ admits an adjacency labeling scheme of size $O(\log n)$. Moreover, these labels can be constructed deterministically.*

Proof. Suppose $\mathcal{F}$ admits an (s,k)-equality based labeling scheme, for constants s,k. For each $G \in \mathcal{F}_n$, we may assume that each equality code $q_i(x)$ is in $[n]$, since there are at most n vertices. Therefore for every x we can assign label $\ell(x) = (p(x), q_1(x), \dots, q_k(x))$ using at most $s + k\lceil \log n \rceil = O(\log n)$ bits. □

The following proposition shows that equality-based labeling must fail to produce $O(\log n)$-size adjacency labeling schemes whenever the family $\mathcal{F}$ is not stable.

Proposition 2.9. *If a hereditary graph family $\mathcal{F}$ admits a constant-size equality-based labeling scheme, then $\mathcal{F}$ is stable.*

Proof. By Lemma 2.7, it holds that $\mathcal{F}$ has a constant-size PUG. By Proposition 1.12 this implies that $\mathcal{F}$ is stable. □

2.5 Basic Adjacency Sketches

Here we give some simple adjacency sketches for equivalence graphs and bounded-arboricity graphs that we will require for our later results.

Definition 2.10. A graph G is an *equivalence graph* if it is a disjoint union of cliques. 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 $i \neq j$.

The equivalence graphs are exactly the P_3-free graphs and the bipartite equivalence graphs are exactly the P_4-free bipartite graphs. The following fact is an easy exercise.

Fact 2.11. *The equivalence graphs and the bipartite equivalence graphs admit constant-size equality-based labeling schemes.*

Definition 2.12. A graph $G = (V, E)$ has *arboricity* α if its edges can be partitioned into at most α forests.

In the next lemma, we interpret the classic labeling scheme of [KNR92] as an equality-based labeling scheme, and we obtain adjacency sketches for bounded-arboricity graphs that improves slightly upon the naïve bound in Lemma 2.7 and in [Har20].

Lemma 2.13. *For any $\alpha \in \mathbb{N}$, let $\mathcal{A}$ be the family 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 $F_1, \dots, F_\alpha$ 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

$$\left(\bigvee_{j=2}^{\alpha} \textsc{Eq}(q_1(x), q_j(y)) \right) \vee \left(\bigvee_{j=2}^{\alpha} \textsc{Eq}(q_1(y), q_j(x)) \right).$$

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 Lemma 2.7 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 $O(\log(\alpha) + \alpha) = O(\alpha)$. □

Remark 2.14. Bounded-arboricity families include many commonly-studied hereditary graph families. Bounded degeneracy families, bounded treewidth families, and proper minor-closed families [Mad67] all have bounded arboricity.

3 Graph Theory & Contextual Results

The hereditary graph families form a lattice, since for any two hereditary families $\mathcal{F}$ and $\mathcal{H}$, it holds that $\mathcal{F} \cap \mathcal{H}$ and $\mathcal{F} \cup \mathcal{H}$ are also hereditary families. In this section we review the structure of this lattice, and give some basic results that place the set of constant-PUG families within this lattice. For an illustrated summary of this section, see Figure 4.

3.1 The Speed of Hereditary Graph Classes

The speed $|\mathcal{F}_n|$ of a hereditary graph family 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 families. Scheinerman & Zito [SZ94] and Alekseev [Ale97] showed that the four smallest *layers* of hereditary graph families are the following:

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

The graph families 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 families of theoretical or practical importance. Despite this, no general characterization is known for them apart from the definition.

3.2 Constant-Size Deterministic Labeling Schemes

This paper asks which subset of the hereditary factorial families correspond to the communication problems with constant-cost randomized protocols. Replacing *randomized* protocols with *deterministic* protocols, we get a question that is quickly answered by the existing literature (this corresponds to the gray-colored areas in Figure 1, Figure 2, and Figure 4). By the argument in Proposition 1.6, these protocols correspond to constant-size (deterministic) adjacency labeling schemes, so our question is answered by a result of Scheinerman [Sch99]: a hereditary family $\mathcal{F}$ admits a constant-size adjacency labeling scheme if and only if it belongs to the *constant, polynomial*, or *exponential* layer. Such families have a bounded number of equivalence classes of vertices, where two vertices x, y are equivalent if their neighborhoods satisfy $N(x) \setminus \{y\} = N(y) \setminus \{x\}$.

The following proposition follows from the arguments in Section 1.1.

Proposition 3.1. *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 family $\mathcal{F}$ is in the constant, polynomial, or exponential layer if and only if there is a constant-cost deterministic protocol for $\textsc{Adj}_{\mathcal{F}}$.*

On the other hand, adjacency labels for a factorial family must have size $\Omega(\log n)$ since graphs in the minimal factorial families 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, similar to the jump in *randomized* label size from $O(1)$ in the *stable* factorial families to $\Omega(\log n)$ for the *unstable* factorial families that would follow from Conjecture 1.2.

3.3 Minimal Factorial Families

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

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

These families were identified by Alekseev [Ale97], and similar results were independently obtained by Balogh, Bollobás, & Weinreich [BBW00].

Each minimal factorial family is either a family of bipartite graphs, or a family of *co-bipartite* graphs (i.e. complements of bipartite graphs), or a family of *split* graphs (i.e. graphs whose vertex set can be partitioned into a clique and an independent set). Six of the minimal families are the following:

- $\mathcal{M}^{\circ\circ}$ is the family of bipartite graphs of degree at most 1.
- $\mathcal{M}^{\bullet\circ}$ is the family of graphs whose vertex set can be partitioned into a clique and an independent set such that ever vertex in each of the parts is adjacent to at most one vertex in the other part.
- $\mathcal{M}^{\bullet\bullet}$ is the family 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 families $\mathcal{M}^{\circ\circ}$, $\mathcal{M}^{\bullet\circ}$, $\mathcal{M}^{\bullet\bullet}$, 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 families motivate our focus on the *stable* factorial families. They are defined as follows (see Figure 3).

Definition 3.2 (Chain-Like Graphs)**.** For any $k \in \mathbb{N}$, the *half-graph* 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 families, which we collectively refer to as *chain-like graphs* ($\mathcal{C}^{\circ\circ}$ is sometimes called the family of *chain graphs*):

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

Proposition 3.3 ([Ale97]). *The minimal factorial families are*

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

It is clear from the definitions that the families $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet}$ are not stable, while the other minimal families are. The following statement is easily proved from Proposition 1.11.

Fact 3.4. $\mathcal{M}^{\circ\circ}, \mathcal{M}^{\bullet\circ}, \mathcal{M}^{\bullet\bullet}, \mathcal{L}^{\circ\circ}, \mathcal{L}^{\bullet\circ}, \mathcal{L}^{\bullet\bullet}$ *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 family $\mathcal{F}$ 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 3.5. *Let* $\mathcal{F}$ *be a hereditary family of graphs. Then* $\mathcal{F}$ *has bounded chain number if and only if* $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet} \not\subseteq \mathcal{F}$*.*

Proof. Let $** \in \{\bullet\bullet, \bullet\circ, \circ\circ\}$ and suppose $\mathcal{C}^{**} \subseteq \mathcal{F}$. By definition, $\mathcal{C}^{**}$ contains H_k^{**} for any $k \in \mathbb{N}$, so $\mathsf{ch}(\mathcal{C}^{**}) = \infty$ and if $\mathcal{C}^{**} \subseteq \mathcal{F}$ then $\mathsf{ch}(\mathcal{F}) \geq \mathsf{ch}(\mathcal{C}^{**}) = \infty$.

Now suppose $\mathcal{C}^{**} \not\subseteq \mathcal{F}$ for every $** \in \{\bullet\bullet, \bullet\circ, \circ\circ\}$. Then for every $** \in \{\bullet\bullet, \bullet\circ, \circ\circ\}$ there is some m^{**} such that all graphs $G \in \mathcal{F}$ are $H^{**}_{m^{**}}$-free. Hence, for $m = \max(m^{\bullet\bullet}, m^{\bullet\circ}, m^{\circ\circ})$, all graphs $G \in \mathcal{F}$ are $\{H_m^{\bullet\bullet}, H_m^{\bullet\circ}, H_m^{\circ\circ}\}$-free.

It was proved in [CS18] that, due to Ramsey's theorem, for every $m \in \mathbb{N}$ there exists a sufficiently large $k = k(m)$ such that any $\{H_m^{\bullet\bullet}, H_m^{\bullet\circ}, H_m^{\circ\circ}\}$-free graph G has $\mathsf{ch}(G) < k$. Hence $\mathsf{ch}(\mathcal{F}) < k$. □

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

3.4 The Bell Numbers Threshold

There is another interesting 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* families above the Bell numbers. However, unlike the factorial layer, the set of minimal families above the Bell numbers is *infinite*, and it has been characterized explicitly [BBW05, ACFL16]. Once again, the families $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}, \mathcal{C}^{\bullet\bullet}$ are minimal. This means that *all* hereditary families below the Bell numbers are stable. Structural properties of these families were given in [BBW00], which we use to prove the following (proof in Appendix A).

Theorem 3.6. *Let* $\mathcal{F}$ *be a hereditary graph family. Then:*

1. *If* $\mathcal{F}$ *is a minimal family 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).*

4 Bipartite Graphs

In this section, we first show that Conjecture 1.2 is equivalent to the special case of it for bipartite graph families.

Lemma 4.1. *If Conjecture 1.2 holds for all hereditary families of bipartite graphs, then it holds for all hereditary families.*

Next, motivated by this equivalence, we establish Conjecture 1.2 for the *monogenic* bipartite graph families.

Theorem 1.15. *Let H be a bipartite graph such that the family of H-free bipartite graphs is factorial. Then any hereditary subfamily $\mathcal{F}$ of the H-free bipartite graphs has a constant-size PUG if and only if $\mathcal{F}$ is stable.*

4.1 Equivalence to Bipartite Graphs

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

Definition 4.2. For any graph $G = (V, E)$ we define the colored bipartite graph $\mathsf{bip}(G) = (V, V^{\triangleleft}, E^{\triangleleft})$, 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

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

Lemma 4.1 follows immediately from the following properties of the map $\mathsf{bip}(G)$.

Proposition 4.3. *Let $\mathcal{F}$ be any hereditary graph family. Then:*

1. *$\mathcal{F}$ has at most factorial speed if and only if $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ has at most factorial speed.*
2. *$\mathcal{F}$ is stable if and only if $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ is stable.*
3. *$\mathcal{F}$ admits a constant-size PUG if and only if $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ admits a constant-size PUG.*
4. *$\mathcal{F}$ admits a constant-size equality-based labeling scheme if and only if $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ admits a constant-size equality-based labeling scheme.*

Proof. **Property 1.** First note that for each n, bip is an injective map from $\mathcal{F}_n$ to the set of colored bipartite graphs (X, Y, E) with $|X| = |Y| = n$. If $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ has at most factorial speed then $|\mathcal{F}_n| = |\mathsf{bip}(\mathcal{F}_n)_{2n}| = 2^{O(n \log n)}$, so $\mathcal{F}$ has at most factorial speed. Now assume $\mathcal{F}$ has at most factorial speed and let $G \in \mathsf{cl}(\mathsf{bip}(\mathcal{F}))_n$. Then there is $F = (V, E) \in \mathcal{F}$ with $\mathsf{bip}(F) = (V, V^{\triangleleft}, E^{\triangleleft}) \in \mathsf{bip}(\mathcal{F})$ such that $G \sqsubset \mathsf{bip}(F)$. Let $Z_1 \subseteq V$ and $Z_2 \subseteq V$ be such that G is isomorphic to the subgraph of $\mathsf{bip}(F)$ induced by $Z_1 \cup Z_2^{\triangleleft}$. Then $G \sqsubset \mathsf{bip}(F[Z_1 \cup Z_2])$. Notice that $F[Z_1 \cup Z_2] \in \mathcal{F}_m$, where $m = |Z_1| + |Z_2| \leq n$. So there is a map from $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))_n$ to $\bigcup_{m \leq n} \mathcal{F}_m$ where for any $H \in \mathcal{F}_m$, there are at most 2^{2m} induced subgraphs of $\mathsf{bip}(H)$, so there are at most 2^{2m} graphs in $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))_n$ that are mapped to H. We therefore have

$$|\mathsf{cl}(\mathsf{bip}(\mathcal{F}))_n| \leq \sum_{m \leq n} 2^{2m} \cdot |\mathcal{F}_m| \leq n 2^{2n} 2^{O(n \log n)} = 2^{O(n \log n)}.$$

Property 2. First suppose that $\mathsf{ch}(\mathcal{F}) = \infty$. For any $k \in \mathbb{N}$, let $G = (V, E) \in \mathcal{F}$ have $\mathsf{ch}(G) = k$, so that there are disjoint sets $\{a_1, \ldots, a_k\}, \{b_1, \ldots, b_k\} \subset V$ such that a_i, b_j are adjacent if and only if $i \leq j$. It then holds that $\{a_1, \ldots, a_k\} \subset V$ and $\{\triangleleft b_1, \ldots, \triangleleft b_k\} \subset V^{\triangleleft}$ are disjoint subsets of vertices in $\mathsf{bip}(G) = (V, V^{\triangleleft}, E^{\triangleleft})$ witnessing $\mathsf{ch}(\mathsf{bip}(G)) \geq k$. Then for any k it holds that $\mathsf{ch}(\mathsf{cl}(\mathsf{bip}(\mathcal{F}))) \geq k$, so $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ is not stable.

Now suppose that $\mathsf{ch}(\mathsf{cl}(\mathsf{bip}(\mathcal{F}))) = \infty$. For any $k \in \mathbb{N}$ there is $H = (X, Y, E_H) \in \mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ with $\mathsf{ch}(H) \geq 2k$, and there is $G \in \mathcal{F}$ such that $H \sqsubset \mathsf{bip}(G)$. Write $G = (V, E_G)$ so that $X \subset V, Y \subset V^{\triangleleft}, E_H \subset E_G^{\triangleleft}$. For $m = 2k$, let $\{a_1, \dots, a_m\} \subseteq X \subseteq V$ and $\{\triangleleft b_1, \dots, \triangleleft b_m\} \subseteq Y \subseteq V^{\triangleleft}$ be sets such that $a_i, \triangleleft b_j$ are adjacent in H if and only if $i \leq j$. If $\{a_1, \dots, a_m\}, \{b_1, \dots, b_m\}$ are disjoint in V then we are done. Otherwise, let t be the smallest number such that $a_t \in \{b_1, \dots, b_m\}$. Consider two cases.

First suppose $t \geq m/2 = k$. Then $\{a_1, \dots, a_k\}, \{b_1, \dots, b_k\}$ are disjoint, and we are done. Next suppose that $t < m/2 = k$, and consider the sets $A = \{a_t, \dots, a_m\}, B = \{b_t, \dots, b_m\}$. We will show that $A \cap B = \emptyset$. By definition, $a_t \in \{b_1, \dots, b_m\}$. Since $a_t, \triangleleft a_t$ are not adjacent in $\mathsf{bip}(G)$ but a_t is adjacent to all elements in $\{\triangleleft b_t, \dots, \triangleleft b_m\}$, it must be that $\triangleleft a_t = \triangleleft b_p$ for some $p < t$.

Suppose for contradiction that $A \cap B \neq \emptyset$. Hence, there is $q > t$ such that $a_q \in B$, i.e. $a_q = b_r$ for some $r \geq t$. Then a_t is adjacent to $\triangleleft b_r$ since $t \leq r$, so a_t is adjacent to a_q in G. Since $a_t = b_p$ it must be that a_q is adjacent to b_p in G, and therefore a_q is adjacent to $\triangleleft b_p$ in $\mathsf{bip}(G)$. This holds only if $q \leq p$, but by definition we have $p < t < q$, a contradiction. Consequently, we have $A \cap B = \emptyset$, and we conclude that $\mathsf{ch}(G) \geq m - t > m/2 = k$.

Property 3 and Property 4. Any adjacency sketch (or equality-based labeling scheme) for $\mathcal{F}$ can be used for $\mathsf{bip}(\mathcal{F})$ by simply appending a single bit to each label that indicate whether the vertex belongs to the left or right part. Any adjacency sketch (or equality-based labeling scheme) for $\mathsf{bip}(\mathcal{F})$ is inherited by $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$.

From any adjacency sketch for $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$, we can construct a sketch for $\mathcal{F}$: for any graph $G \in \mathcal{F}$ and any vertex $x \in V(G)$, assign the sketch $(\mathsf{sk}(x), \mathsf{sk}(x'))$, where x' is the copy of x in the bipartite transformation. For any pair x, y, the decoder outputs what the bipartite decoder would output given $\mathsf{sk}(x), \mathsf{sk}(y')$.

From any equality-based labeling scheme for $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$, we can construct one for $\mathcal{F}$ similarly. For any graph $G = (V, E) \in \mathcal{F}$, write $\mathsf{bip}(G) = (V, V^{\triangleleft}, E^{\triangleleft})$. To any vertex $x \in V$ we assign the label $(p(x), p(\triangleleft x) \mid q(x), q(\triangleleft x))$ from the labeling scheme for $\mathsf{bip}(G)$. On labels $(p(x), p(\triangleleft x) \mid q(x), q(\triangleleft x))$ and $(p(y), p(\triangleleft y) \mid q(y), q(\triangleleft y))$, simulate the decoder on $(p(x) \mid q(x))$ and $(p(\triangleleft y) \mid q(\triangleleft y))$. ☐

4.2 Decomposition Scheme for Bipartite Graphs

In this section we define a decomposition scheme for bipartite graphs that we will use to establish Conjecture 1.2 for the monogenic bipartite graph families.

Definition 4.4 ($(\mathcal{Q}, k)$-decomposition tree)**.** Let $G = (X, Y, E)$ be a bipartite graph, $k \geq 2$, and let $\mathcal{Q}$ be a hereditary family 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 $\overline{\overline{G'}}$ is disconnected. There are sets $X'_1, \dots, X'_t \subseteq X'$ and $Y'_1, \dots, Y'_t \subseteq Y'$ such that $\overline{\overline{G[X'_1, Y'_1]}}, \dots, \overline{\overline{G[X'_t, Y'_t]}}$ are the connected components of $\overline{\overline{G'}}$. 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 $X'_1, X'_2, \ldots, X'_p \subseteq X'$ and $Y'_1, Y'_2, \ldots, Y'_q \subseteq Y'$, where $p \le k$, $q \le k$. 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 $X'_1, X'_2, \ldots, X'_p$ and $Y'_1, Y'_2, \ldots, Y'_q$.

Lemma 4.5. *Let $k \ge 2$ and $d \ge 1$ be natural constants, and let $\mathcal{Q}$ be a family of bipartite graphs that admits a constant-size equality-based adjacency labeling scheme. Let $\mathcal{F}$ be a family 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 2.6), 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 $X'_1, \ldots, X'_t \subseteq X', Y'_1, \ldots, Y'_t \subseteq Y$ such that the children $v_1, \ldots, v_t$ are the connected components $G_v[X'_1, Y'_1], \ldots, G_v[X'_t, Y'_t]$ 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 $\overline{\overline{G_v}}$ is disconnected, with sets $X'_1, \ldots, X'_t \subseteq X', Y'_1, \ldots, Y'_t \subseteq Y$ such that $\overline{\overline{G_v[X'_1, Y'_1]}}, \ldots, \overline{\overline{G_v[X'_t, Y'_t]}}$ are the connected components of $\overline{\overline{G_v}}$, and the children $v_1, \ldots, v_t$ of v are the graphs $G_v[X'_1, Y'_1], \ldots, G_v[X'_t, Y'_t]$. We define $\ell_v(x) = (\overline{D} \mid j), \ell_{v_j}(x)$, 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 $X'_1, \ldots, X'_p \subseteq X', Y'_1, \ldots, Y'_q \subseteq Y'$ be the partitions of X', Y' with $p, q \le k$. 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 $\ell_v(x) = (P \mid i, q), \ell_{v_{i,1}}(x), \ldots, \ell_{v_{i,q}}(x)$, where $\ell_{v_{i,j}}(x)$ is the label assigned to x at node $v_{i,j}$. If $x \in Y$, then we define $\ell_v(x) = (P \mid i, p), \ell_{v_{1,i}}(x), \ldots, \ell_{v_{p,i}}(x)$, 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, $f(i) \le 1 + k \cdot f(i+1)$ and $f(d-1) = 1$, which implies that the total number of tuples in $\ell(x)$ does not exceed $f(0) \le k^d$. 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 $(\alpha(x) \mid -)$ and $(\alpha(y) \mid -)$ 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 $(L \mid -), \ell'_v(x)$ and $(L \mid -), \ell'_v(y)$, 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 $(D \mid i), \ell_{v_i}(x)$ and $(D \mid j), \ell_{v_j}(y)$, 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 $(\overline{D} \mid i), \ell_{v_i}(x)$ and $(\overline{D} \mid j), \ell_{v_j}(y)$, 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 $(P \mid i, q), \ell_{v_{i,1}}(x), \ldots, \ell_{v_{i,q}}(x)$ and $(P \mid j, p), \ell_{v_{1,j}}(y), \ldots, \ell_{v_{p,j}}(y)$ 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 4.6 (($\mathcal{Q}, k$)-tree for general graphs)**.** A similar decomposition scheme can be used for non-bipartite graph families; we do this for permutation graphs in Section 5.2.

4.3 Monogenic Bipartite Graph Families

Let $\mathcal{H}$ be a *finite* set of bipartite graphs. It is known [All09] that if the family 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 families of bipartite graphs. More specifically, it was shown that, for a colored bipartite graph H, the family 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.

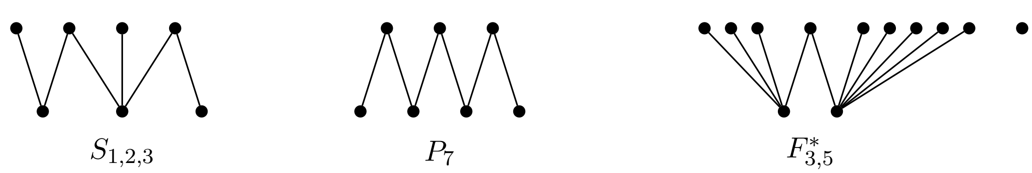

Figure 5: The bipartite graphs from Definition 4.7

Definition 4.7 ($S_{1,2,3}$, P_7, $F^*_{p,q}$)**.** See Figure 5 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 4.8 ([All09, LZ17]). *Let H be a colored bipartite graph, and let $\mathcal{F}$ be the family of H-free bipartite graphs. If $\mathcal{F}$ has at most factorial speed, then $\mathcal{F}$ is a subfamily 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 Conjecture 1.2 for monogenic families of bipartite graphs (i.e. to prove Theorem 1.15), it is enough to consider the *maximal* monogenic factorial families of bipartite graphs defined by $S_{1,2,3}$, P_7, $F^*_{p,q}$.

4.3.1 $S_{1,2,3}$-Free Bipartite Graphs

In this section, we derive Theorem 1.15 for the family of $S_{1,2,3}$-free bipartite graphs. It is known that the family 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 families of bounded twin-width (Theorem 1.19).

Theorem 4.9. *Let $\mathcal{F}$ be a stable family of $S_{1,2,3}$-free bipartite graphs. Then $\mathcal{F}$ admits a constant-size equality-based adjacency labeling scheme, and hence $\mathsf{SK}(\mathcal{F}_n) = O(1)$.*

4.3.2 $F^*_{p,q}$-Free Bipartite Graphs

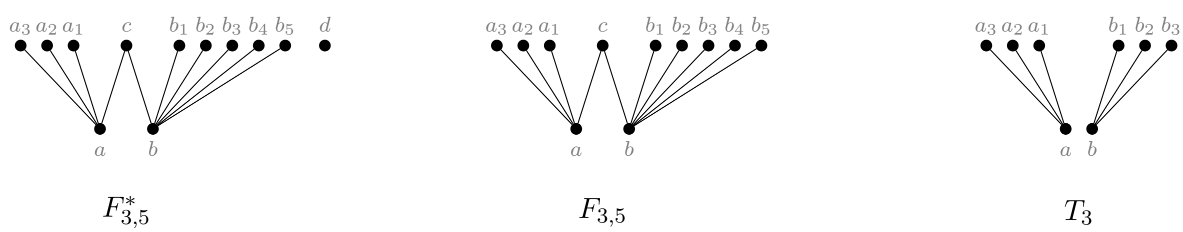

Figure 6: The bipartite graphs considered in Section 4.3.2.

In this section, we prove Theorem 1.15 for families of $F^*_{p,q}$-free bipartite graphs by developing a constant-size equality-based adjacency labeling scheme for stable families of $F^*_{p,q}$-free bipartite graphs via a sequence of labeling schemes for special subfamilies 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 $(a, c), (b, c), \{(a, a_i) \mid i \in [p]\}, \{(b, b_i) \mid j \in [q]\}$. We also denote by T_p the bipartite graph on 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 4.10. 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 $Y = Y_1 \cup \ldots \cup Y_q$ each of size s, and for every $i \in [q]$:

(1) x_i is adjacent to all vertices in Y_j for all $1 \le j \le i$, and

(2) x_i is adjacent to no vertices in Y_j for all $i < j \le q$.

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 and 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 4.11. *Let $G=(X,Y,E)$ be any one-sided T_p-free bipartite graph and let $u,v \in X$ satisfy* $\deg(u) \le \deg(v)$. *Then $|N(u)\cap N(v)| > |N(u)| - p$.*

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

Proposition 4.12. *Suppose $S_1,\dots,S_t \subseteq [n]$ each have $|S_i| \ge n-p$ where $n > pt$. Then $\left|\bigcap_{j=1}^t S_j\right| \ge n-pt$.*

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

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

so $\left|\bigcap_{j=1}^t S_j\right| \ge n-|R| \ge n-pt$. □

Lemma 4.13. *Fix any constants k,q,p such that $k \ge qp+1$ and let $G=(X,Y,E)$ be any one-sided T_p-free bipartite graph. Then there exists $m \ge 0$ and partitions $X = A_0 \cup A_1 \cup \ldots \cup A_m$ and $Y = B_1 \cup \ldots \cup B_m \cup B_{m+1}$, where $A_i \ne \emptyset$, $B_i \ne \emptyset$ for every $i \in [m]$, such that the following hold*

(1) *$|B_i| \ge k$, 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 $\bigcup_{i\ge j+1} B_i$.*

(3) *For every $i,j, 1 \le i \le j \le m$, every $x \in A_j$ has more than $|B_i|-p$ neighbours in B_i.*

(4) *If $m \ge q$, then $Z_{q,k-qp}$ 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 4.11.

Finally, to prove condition (4) we will show that for any $q \leq m$ there exist sets $B'_1 \subseteq B_1, \ldots, B'_q \subseteq B_q$ so that the vertices $\{a_1, \ldots, a_q\}$ and the sets $B'_1, \ldots, B'_q$ 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 4.12 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 $\{a_1, \ldots, a_q\}$ and the sets $B'_1, \ldots, B'_q$ induce $Z_{q,k-pq}$, which proves condition (4) and concludes the proof of the lemma. ☐

Lemma 4.14. *Let $p \in \mathbb{N}$ and let $\mathcal{T}$ be a stable family of one-sided T_p-free bipartite graphs. Then $\mathcal{T}$ admits a constant-size equality-based adjacency labeling scheme, and hence $\mathsf{SK}(\mathcal{T}_n) = O(1)$.*

Proof. Since $\mathcal{T}$ is stable, it does not contain $\mathcal{C}^{\circ\circ}$ as a subfamily. 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 $X = A_0 \cup A_1 \cup \ldots \cup A_m$ and $Y = B_1 \cup \ldots \cup B_m \cup B_{m+1}$ be partitions satisfying the conditions of Lemma 4.13. 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 $i \in \{0, 1, \ldots, m\}$ is the unique index such that $x \in A_i$. This tuple follows by i tuples $(- \mid y_1^j, y_2^j, \ldots, y_{p_j}^j)$, $j \in [i]$, where $p_j < p$ and $\{y_1^j, y_2^j, \ldots, y_{p_j}^j\}$ are the non-neighbours of x in B_j. The last tuple of $\ell(x)$ is $(- \mid y_1^{i+1}, y_2^{i+1}, \ldots, y_{k'}^{i+1})$, where $k' < k$ and $y_1^{i+1}, y_2^{i+1}, \ldots, y_{k'}^{i+1}$ 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 $y_1^j, y_2^j, \ldots, y_{p_j}^j$ 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 $y_1^{i+1}, y_2^{i+1}, \ldots, y_{k'}^{i+1}$ 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. ☐

Next, we develop an equality-based labeling scheme for stable families 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 4.15. *Let $G = (X, Y, E)$ be any one-sided $F_{p,p}$-free bipartite graph and let $u, v \in X$ satisfy $\deg(u) \le \deg(v)$. 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 $|N(u) \setminus N(v)| \ge p$. Since $\deg(u) \le \deg(u)$, this means $|N(v) \setminus N(u)| \ge |N(u) \setminus N(v)| \ge p$. 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. □

Proposition 4.16. *Let $G = (X, Y, E)$ be any one-sided $F_{p,p}$-free bipartite graph and let $x, y, z \in X$ satisfy $\deg(x) \ge \deg(y) \ge \deg(z) \ge 2p$. 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 $|N(y) \cap N(z)| > |N(z)| - p \ge p$ by Proposition 4.15.

Suppose that $N(x) \cap N(y) \neq \emptyset$. For contradiction, assume that $N(x) \cap N(y) \cap N(z) = \emptyset$. Then $|N(y) \setminus N(x)| \ge |N(y) \cap N(z)| > |N(z)| - p \ge p$, 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 $|N(x) \cap N(z)| \le |N(z) \setminus N(y)| < p \le |N(z)| - p < |N(x) \cap N(z)|$, a contradiction. □

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 4.17. *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 4.16 and 4.15, 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 $N(y_{t-1}) \cap N(x) = N(y_{t-1}) \cap N(y_t) \neq \emptyset$. □

Lemma 4.18. *Fix any constants $p, q \ge 1$, 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 $p < s \le k$, 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 then 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 4.17, 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 $u_1, \ldots, u_r \in X_2$ and pairwise disjoint sets $V_1 \subseteq N(u_1), \ldots, V_r \subseteq N(u_r)$ 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 $u_1, \ldots, u_r$ 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 4.17, 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 4.17. 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 $\{u_1, \ldots, u_r, x\} \cup V_1' \cup V_2' \cup \ldots \cup V_r' \cup V_{r+1}'$ contains $Z_{r+1, s-p}$ as an induced subgraph. □

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 family 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 4.18, 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 4.18. 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 4.19. *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 $\mathcal{P} = (v_0, v_1, v_2, \ldots, v_s)$ 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 $X^i = X^i_0 \cup X^i_1 \cup X^i_2$ be the partition of X^i according to the decomposition rules. Since $X^i_0 \cup X^i_1 \neq \emptyset$ and $X^i_0 = \emptyset$, we conclude that $X^i_1 \neq \emptyset$. Therefore, by Lemma 4.18, if the $G_{v_{i-1}} = G[X^i_2, Y^i]$ 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 $H^{\circ\circ}_q = Z_{q,1} \sqsubset Z_{q,k-(q-1)p} \sqsubset G_{v_{t+2q-3}} \sqsubset G$, a contradiction. □

Lemma 4.20. *Let $p \in \mathbb{N}$ and let $\mathcal{F}$ be a stable family of one-sided $F_{p,p}$-free bipartite graphs. Then $\mathcal{F}$ admits a constant-size equality-based adjacency labeling scheme, and hence $\mathsf{SK}(\mathcal{F}_n) = O(1)$.*

Proof. Since $\mathcal{F}$ is stable, it does not contain $\mathcal{C}^{\circ\circ}$ as a subfamily. Let q be the minimum number such that $H^{\circ\circ}_q \notin \mathcal{F}$. 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 4.19, the graphs in $\mathcal{F}$ admit $(\mathcal{Q}, 2)$-decomposition trees of depth at most $2q$. Hence, by Lemma 4.14 and Lemma 4.5, $\mathcal{F}$ admits a constant-size equality-based adjacency labeling scheme. □

We conclude this section by showing that stable families 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 4.21 ([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 $\overline{G[X_2, Y_1]}$ are one-sided $F_{p,p}$-free.*

Theorem 4.22. *For any constants $p, p' \geq 1$, a stable family $\mathcal{F}$ of $F^*_{p,p'}$-free bipartite graphs admits a constant-size equality-based adjacency labeling scheme, and hence $\mathsf{SK}(\mathcal{F}_n) = O(1)$.*

Proof. As before, since $\mathcal{F}$ is stable, it does not contain $\mathcal{C}^{\circ\circ}$ as a subfamily. Let q be the minimum number such that $H^{\circ\circ}_q \notin \mathcal{F}$, and assume without loss of generality that $p \geq p'$. It follows that $\mathcal{F}$ is a subfamily of $(F^*_{p,p}, H^{\circ\circ}_q)$-free bipartite graphs. Let $G = (X, Y, E)$ be a member of this family. Let $X = X_1 \cup X_2, Y = Y_1 \cup Y_2$ be the partition given by Proposition 4.21. 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 4.20 for $G[X_1, Y_1]$. If $x \in X_2$, complete the label by using the labeling scheme of Lemma 4.20 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 4.20 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. □

4.3.3 P_7-Free Bipartite Graphs

In this section, we prove Theorem 1.15 for P_7-free bipartite graphs by developing a constant-size equality-based adjacency labeling scheme for stable families 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 4.23 (Chain Decomposition)**.** See Figure 7 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 $i \le k-1$, 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 $j \ge i$;
  - the set C_i is complete to B_j for $j < i$, and is anticomplete to B_j for $j \ge i$.

Remark 4.24. 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 4.25 ([LZ17])**.** *Let $G = (X, Y, E)$ be a P_7-free bipartite graph such that both G and $\overline{\overline{G}}$ are connected. Then G or $\overline{\overline{G}}$ admits a k-chain decomposition for some $k \ge 2$.*

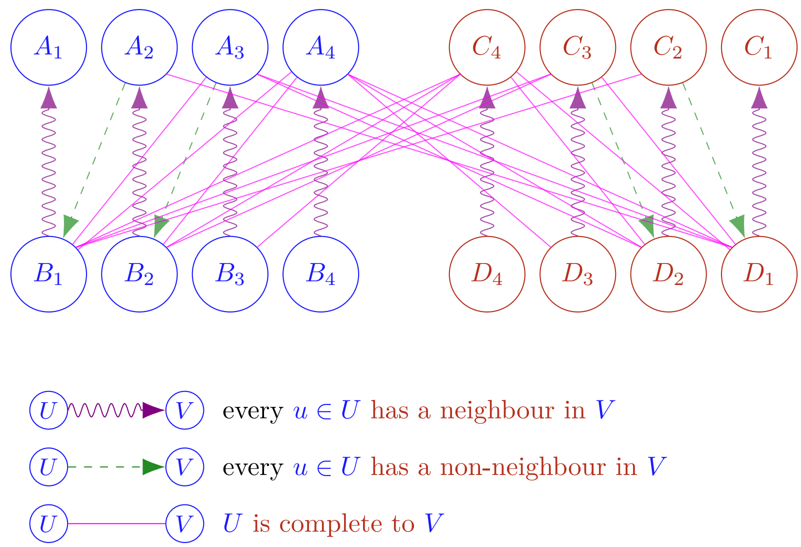

Figure 7: Example of a 4-chain decomposition.

Lemma 4.26. *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 $X_1, X_2, \ldots, X_p$, and a partition of Y into $q \leq 2(c+1)$ sets $Y_1, Y_2, \ldots, Y_q$ such that, for any $i \in [p], j \in [q]$,*

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

Proof. Assume, without loss of generality, that X is partitioned into subsets $A_1, \ldots, A_k, C_1, \ldots, C_k$ and Y is partitioned into subsets $B_1, \ldots, B_k, D_1, \ldots, D_k$ satisfying Definition 4.23. 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 $a_2 \in A_2, a_3 \in A_3, \ldots, a_k \in A_k$, and $d_1 \in D_1, d_2 \in D_2, \ldots, d_{k-1} \in D_{k-1}$ the subgraph of G induced by $\{a_2, a_3, \ldots, a_k, d_1, d_2, \ldots, d_{k-1}\}$ is isomorphic to $H^{\circ\circ}_{k-1}$, 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 $H^{\circ\circ}_2$ and $\mathsf{ch}(G) \geq 2$. We split our analysis in two cases.

Case 1. $k \geq 3$. We will show that for any $X' \in \{A_1, \ldots, A_k, C_1, \ldots, C_k\}$ and $Y' \in \{B_1, \ldots, B_k, D_1, \ldots, D_k\}$, $\mathsf{ch}(G[X', Y']) < \mathsf{ch}(G)$. Since $\mathsf{ch}(G) \geq 2$, 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 4.23, 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 $i \in \{2, \ldots, k\}$.

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 $\mathsf{ch}(G[A_1, B_1]) < \mathsf{ch}(G[A_1 \cup \{c_2\}, B_1 \cup \{d_2\}]) \leq \mathsf{ch}(G)$. Similarly, since D_1 is complete to all $A_2, A_3, \ldots, A_k$, and C_1 is anticomplete to all $B_1, B_2, \ldots, B_k$, 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 $i \in \{2, \ldots, k\}$ 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}), $i \in \{2, \ldots, k\}$.

In this case, $A_1, \ldots, A_k, C_1, \ldots, C_k$ and $B_1, \ldots, B_k, D_1, \ldots, D_k$ 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 4.24), 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, $\mathsf{ch}(G[A_1, B_1]) < \mathsf{ch}(G[A_1 \cup \{c_2\}, B_1 \cup \{d_1\}]) \leq \mathsf{ch}(G)$. Similarly, because C_1 is anticomplete to both B_1 and B_2 and D_1 is complete to A_2, we have that $\mathsf{ch}(G[A_2, B_1]) < \mathsf{ch}(G[A_2 \cup \{c_1\}, B_1 \cup \{d_1\}]) \leq \mathsf{ch}(G)$ and $\mathsf{ch}(G[A_2, B_2]) < \mathsf{ch}(G[A_2 \cup \{c_1\}, B_2 \cup \{d_1\}]) \leq \mathsf{ch}(G)$. 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 $\mathsf{ch}(G[X', Y']) < \mathsf{ch}(G)$. 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 $\mathsf{ch}(G[A_1, B_1']) < \mathsf{ch}(G[A_1 \cup \{a_2\}, B_1' \cup \{d_1\}]) \leq \mathsf{ch}(G)$.

To establish the desired property for (A_1, B_1''), we first observe that by Remark 4.24 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 $\mathsf{ch}(G[A_1, B_1'']) < \mathsf{ch}(G)$. □

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 4.27. *Let $G = G(X, Y, E)$ be a P_7-free bipartite graph such that both G and $\overline{\overline{G}}$ 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 $X_1, X_2, \ldots, X_p$, and a partition of Y into $q \leq 2(c+2)$ sets $Y_1, Y_2, \ldots, Y_q$ such that for any $i \in [p]$, $j \in [q]$*

$$\Big(\mathsf{ch}(G_{i,j}), \mathsf{ch}(\overline{\overline{G_{i,j}}})\Big) \prec \Big(\mathsf{ch}(G), \mathsf{ch}(\overline{\overline{G}})\Big),$$

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

Proof. It is easy to verify that for any $k \geq 2$, the graph $\overline{\overline{H_k^{\circ\circ}}}$ contains the half graph $H_{k-1}^{\circ\circ}$, which implies that $\mathsf{ch}(\overline{\overline{G}}) \leq \mathsf{ch}(G) + 1 = c + 1$. 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 4.25, G or $\overline{\overline{G}}$ admits a k-chain decomposition for some $k \geq 2$. Therefore, by Lemma 4.26 applied to either G or $\overline{\overline{G}}$, there exist a partition of X into at most $p \leq 2(c+2)$ sets $X_1, X_2, \ldots, X_p$, and a partition of Y into at most $q \leq 2(c+2)$ sets $Y_1, Y_2, \ldots, Y_q$ such that either $\mathsf{ch}(G_{i,j}) < \mathsf{ch}(G)$ holds for any $i \in [p]$, $j \in [q]$, or $\mathsf{ch}(\overline{\overline{G_{i,j}}}) < \mathsf{ch}(\overline{\overline{G}})$ 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. □

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 family 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 $\overline{\overline{G_v}}$ is disconnected, then v is a $\overline{D}$-node. There are sets $X'_1, \ldots, X'_t \subseteq X'$ and $Y'_1, \ldots, Y'_t \subseteq Y'$ such that $\overline{\overline{G[X'_1, Y'_1]}}, \ldots, \overline{\overline{G[X'_t, Y'_t]}}$ are the connected components of $\overline{\overline{G_v}}$. The children of this node are $G[X'_1, Y'_1], \ldots, G[X'_t, Y'_t]$.
- If G_v does not belong to $\mathcal{Q}$, and neither G_v, nor $\overline{\overline{G_v}}$ is disconnected, then v is a P-node. Let $X'_1, X_2, \ldots, X'_p$ be a partition of X' into $p \leq 2(c+2)$ sets, and $Y'_1, Y_2, \ldots, Y'_q$ be a partition of Y' into $q \leq 2(c+2)$ sets, as in Lemma 4.27. The children of this node are $G[X'_i, Y'_j]$, $i \in [p]$, $j \in [q]$.

Claim 4.28. *Let $\mathcal{T}$ be a decomposition tree as a 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 $\mathsf{ch}(G_1) < \mathsf{ch}(G_3)$.*

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, $\mathsf{ch}(G[X_1, Y_1]) < \mathsf{ch}(G[\{x'\} \cup X_1, \{y'\} \cup Y_1]) \leq \mathsf{ch}(G_3)$. If both sets X'_1 and Y'_1 are non-empty, the argument is similar and we omit the details. ☐

Theorem 4.29. *Let $\mathcal{F}$ be a stable family of P_7-free bipartite graphs. Then $\mathcal{F}$ admits a constant-size equality-based adjacency labeling scheme, and hence $\mathsf{SK}(\mathcal{F}_n) = O(1)$.*

Proof. Since $\mathcal{F}$ is stable, it does not contain $\mathcal{C}^{\circ\circ}$ as a subfamily. 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 family 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 4.27, 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 $\left(\mathsf{ch}(G'), \mathsf{ch}(\overline{\overline{G'}})\right)$ and we will prove that if the length of the path from the root G to a node G' is at least $6c$, then $\mathsf{ch}(G') \leq 1$ or $\mathsf{ch}(\overline{\overline{G'}}) \leq 1$, 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 4.28 (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 4.27, for the child node H' of the P-node H on the path $\mathcal{P}$, we have $\left(\mathsf{ch}(H'), \mathsf{ch}(\overline{\overline{H'}})\right) \prec \left(\mathsf{ch}(H), \mathsf{ch}(\overline{\overline{H}})\right)$. In the latter case, by Claim 4.28 (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 $\left(\mathsf{ch}(H'), \mathsf{ch}(\overline{\overline{H'}})\right) \prec \left(\mathsf{ch}(H), \mathsf{ch}(\overline{\overline{H}})\right)$.

Now, since for the root node G we have $\left(\mathsf{ch}(G), \mathsf{ch}(\overline{\overline{G}})\right) \prec (c, c+1)$, if $\mathcal{P}$ has length at least $6c$, then $\mathsf{ch}(G') \leq 1$ or $\mathsf{ch}(\overline{\overline{G'}}) \leq 1$, as required. The result now follows from Lemma 4.5 and a simple observation that class $\mathcal{Q}$ admits a constant-size equality-based adjacency labeling scheme. □

5 Interval & Permutation Graphs

In the following section we confirm Conjecture 1.2 for subfamilies of interval and permutation graphs. Interval graphs are the intersection graphs of intervals on the real line, and are arguably the most studied class of geometric intersection graphs. As pointed out in [Har20], interval graphs contain the co-chain graphs $\mathcal{C}^{\bullet\bullet}$ and hence cannot admit a constant-size PUG, so Conjecture 1.2 holds for the family of interval graphs. Here we wish to show that Conjecture 1.2 also holds for all subfamilies of interval graphs.

Definition 5.1 (Interval graph)**.** A graph G is an *interval graph* if there exists an *interval realization* $\ell, r : V(G) \to \mathbb{R}$ 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 family of interval graphs is a factorial family and it 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 factorial family of geometric intersection graphs (like interval graphs, they are a subfamily of segment intersection graphs). They admit a straightforward $O(\log n)$-bit adjacency labeling scheme that follows from their definition, and so are positive examples to the IGC.

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 $\pi(i) > \pi(j)$.

We confirm our conjecture for all subfamilies of interval and permutation graphs.

Theorem 1.16. *Let $\mathcal{F}$ be any hereditary subfamily of interval or permutation graphs. Then $\mathcal{F}$ admits a constant-size PUG if and only if $\mathcal{F}$ is stable.*

This theorem follows from Theorem 5.7 (interval graphs) and Theorem 5.20 (permutation graphs), proved below.

5.1 Interval Graphs

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

Lemma 5.3. *Let $\mathcal{F}$ be a family 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 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 2.13, $\mathcal{F}$ admits a constant-size equality-based labeling scheme and an adjacency sketch of size $O(c)$. □

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 5.4. 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 5.5. *Let $\mathcal{F}$ be any hereditary graph family 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. □

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

Lemma 5.6. *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 $\ell, r : V(G) \to \mathbb{R}$ 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 $X = \{x_1, \ldots, x_c\}$ be the c vertices that form a c-clique in G, arranged so that $\ell(x_1) < \ell(x_2) < \cdots < \ell(x_c)$. Consider the sequence $r = (r(x_1), \ldots, r(x_c))$ 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 $y_1, \ldots, y_k$ with $\ell(y_1) < \cdots < \ell(y_k)$ and either $r(y_1) < \cdots < r(y_k)$ or $r(y_1) > \cdots > r(y_k)$. Graphically speaking, the intervals for $y_1, \ldots, y_k$ 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, $r(y_1) < \cdots < r(y_k)$; see Figure 8. The pyramid case is similar.

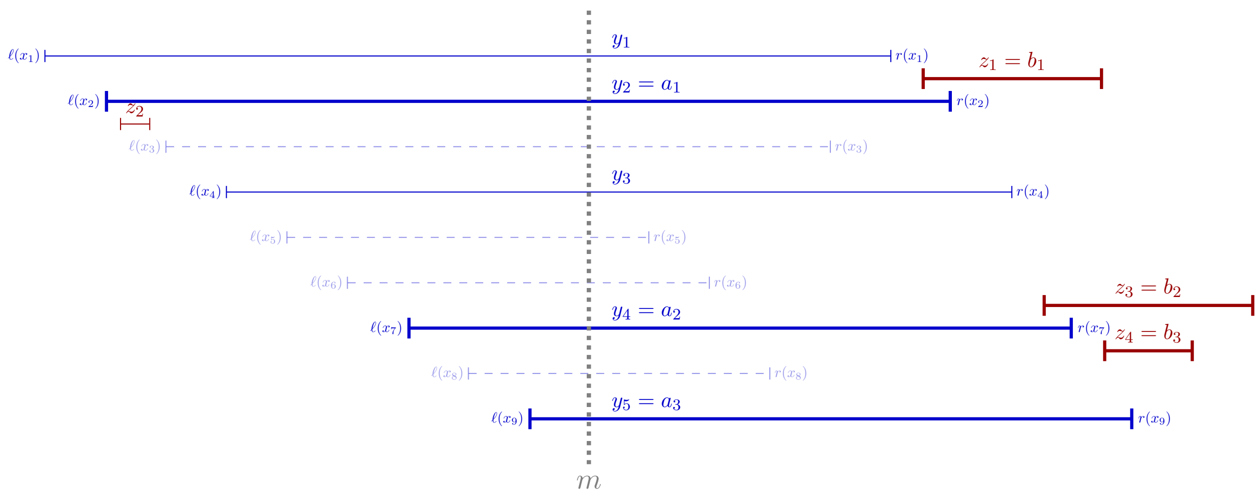

Figure 8: Example illustrating the notation from the proof of Lemma 5.6. The fat blue and red intervals, $a_1, \ldots, a_3$ resp. $b_1, \ldots, b_3$, form an induced subgraph with chain number 3.

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 $i(z_1), \ldots, i(z_{k-1})$, at least $h = \lceil (k-1)/2 \rceil$ are one the same side of m. Assume the majority is on the right; the other case is similar. So for $1 \leq j_1 < \cdots < j_h \leq k-1$ intervals $i(z_{j_1}), \ldots, i(z_{j_h})$ are all to the right of m. Define $B = (b_1, \ldots, b_h) = (z_{j_1}, \ldots, z_{j_h})$ and

$A = (a_1, \ldots, a_h) = (y_{j_1+1}, \ldots, y_{j_h+1})$. By definition, $\ell(b_1) < r(a_1) < \ell(b_2) < r(a_2) < \cdots < \ell(b_h) < r(a_h)$, so a_i is adjacent to b_j if and only if $j \leq i$. Hence $G[A, B]$ is isomorphic to $H_h^{\circ\circ}$. It is easy to check that $h = \lceil (\lfloor \sqrt{c} \rfloor - 1)/2 \rceil = \lfloor \sqrt{c}/2 \rfloor$. □

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

Theorem 5.7. *Let $\mathcal{F}$ be a stable family of interval graphs. Then $\mathcal{F}$ admits a constant-size equality-based adjacency labeling scheme, and hence $\mathsf{SK}(\mathcal{F}_n) = O(1)$.*

Proof. Since $\mathcal{F}$ is stable, we have $\mathsf{ch}(\mathcal{F}) = k$ 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 $\mathsf{ch}(G') \leq k$, and hence the clique number of G' is at most $4(k+1)^2$ by (contraposition of) Lemma 5.6. So $\mathcal{F}'$ is a family of interval graphs with clique number bounded by $4(k+1)^2$, and hence by Lemma 5.3, it admits a constant-size equality-based labeling scheme (and a size $O(k^2)$ adjacency sketch). By Lemma 5.5, so does $\mathcal{F}$. □

Remark 5.8. 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 5.5, but does not recover this explicit bound on the sketch size. We prove in Section 8 that any stable family $\mathcal{F}$ with bounded twin-width admits a constant-size equality-based labeling scheme. Although interval graphs do not have bounded twin-width, some subfamilies of interval graphs (e.g. unit interval graphs) are known to have bounded twin-width [BKTW20]. We claim that any stable family of interval graphs has bounded twin-width. A sketch of this proof is as follows.

Proposition 5.9. *Let $\mathcal{F}$ be a stable family of interval graphs. Then $\mathcal{F}$ has bounded twin-width.*

Proof sketch. As in Lemma 5.5, for any graph $G \in \mathcal{F}$ that is not twin-free, we define equivalence classes $T_1, \ldots, T_m \subseteq V(G)$ of vertices equivalent under the twin relation (instead of the true-twin or false-twin relation). By choosing arbitrary representatives $t_i \in T_i$ and letting $T = \{t_1, \ldots, t_m\}$, we obtain a graph $G[T]$. This graph is *not* necessarily twin-free; however, $G[T]$ has fewer vertices than G. Repeat this operation until it terminates at a twin-free graph $G[T']$.

It holds that the twin-width satisfies $\mathsf{tww}(G[T']) = \mathsf{tww}(G)$, which is straightforward to check from Definition 8.1. By Lemma 5.6, it holds that the graphs $G[T']$ obtained in this way have bounded clique number. Then, by the argument in the proof of Lemma 5.3, they have bounded treewidth, and therefore bounded twin-width. Since $\mathsf{tww}(G) = \mathsf{tww}(G[T'])$ we conclude that $\mathcal{F}$ has bounded twin-width. □

5.2 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 5.10. *For any permutation graph G there is an injective mapping $\phi : V(G) \to \mathbb{R}^2$ 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 $i \in [2]$.*

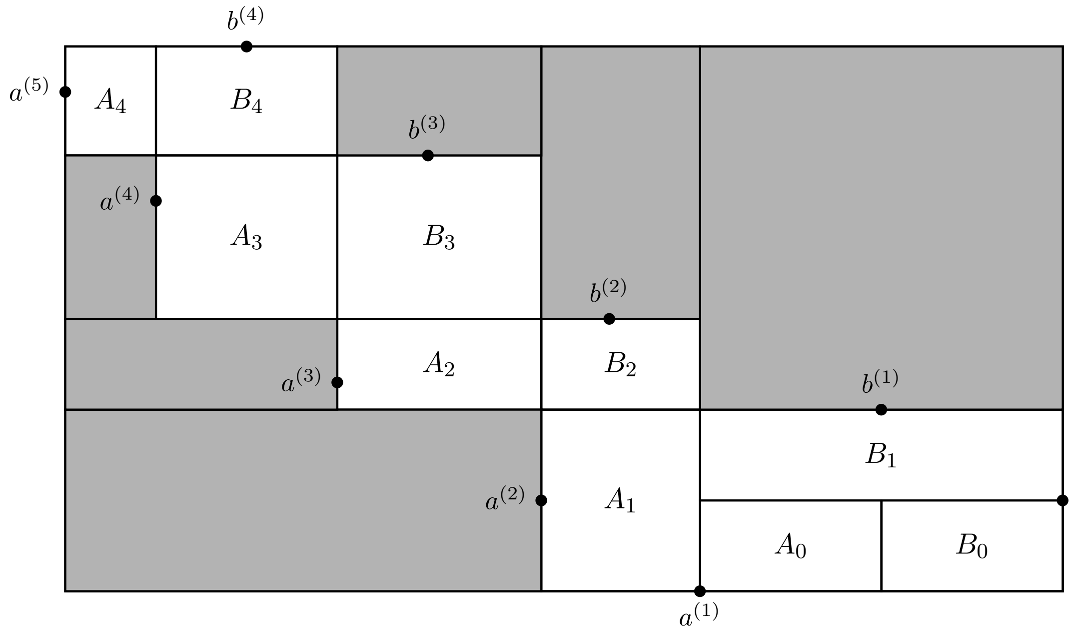

Figure 9: The permutation graph decomposition.

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 5.11. *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 $\mathsf{ch}(G[A \cup \{u, v\}]) > \mathsf{ch}(G[A])$.*

Proof. Suppose $\mathsf{ch}(G[A]) = k$ and let $\{a_1, \ldots, a_k, b_1, \ldots, b_k\}$ be the vertices such that a_i, b_j are adjacent if and only if $i \le j$. Then set $a_{k+1} = u$ and $b_{k+1} = v$, and verify that vertices $\{a_1, \ldots, a_{k+1}, b_1, \ldots, b_{k+1}\}$ satisfy Definition 1.10, so $\mathsf{ch}(G[A \cup \{u, v\}]) \ge k + 1 > \mathsf{ch}(G[A])$. □

A bipartite graph $G = (X, Y, E)$ is called a *chain graph* if it belongs to $\mathcal{C}^{\circ\circ}$. Chain graphs are exactly the $2K_2$-free bipartite graphs, where $2K_2$ is the disjoint union of two edges.

Proposition 5.12. *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 $V_1(-\infty, t) = \{a^{(1)}, \ldots, a^{(s)}\}$ and $V_1(t, \infty) = \{b^{(1)}, \ldots, b^{(t)}\}$ 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 $a_2^{(i)} \le b_2^{(j)}$. 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 $a_2^{(i_2)} \le b_2^{(j_2)}$, 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. □

Any subfamily of $\mathcal{C}^{\circ\circ}$ has a constant-size adjacency labeling scheme, because $\mathcal{C}^{\circ\circ}$ is a minimal factorial family. 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 5.13. *Let $\mathcal{F} \subset \mathcal{C}^{\circ\circ}$ be a hereditary family 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 $A_1, \dots, A_p \subset [n]$ 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 $p, q \le k+1$. Suppose for contradiction that $p \ge k+2$. Since $A_1, \dots, A_p$ are maximal, there exist $b_1, \dots, b_{p-1} \in B$ such that $a_1 < b_1 < a_2 < b_2 < \cdots < b_{p-1} < a_p$, 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 $\mathsf{ch}(G) \ge p-1 \ge k+1$, a contradiction. A similar proof shows that $q \le k+1$.

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 $i \le j$. 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 $i \le j$. □

Definition 5.14 (Permutation Graph Decomposition)**.** 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:

$$\text{For } 2 \le i \le \alpha, \text{ define } A_i := \{a^{(i+1)}\} \cup \left(V_1(a_1^{(i+1)}, a_1^{(i)}) \cap V_2(b_2^{(i-1)}, b_2^{(i)})\right)$$
$$\text{For } 2 \le i \le \beta, \text{ define } B_i := \{b^{(i)}\} \cup \left(V_1(a_1^{(i)}, a_1^{(i-1)}) \cap V_2(b_2^{(i-1)}, b_2^{(i)})\right)$$
$$A_1 := \{a^{(2)}\} \cup \left(V_1(a_1^{(2)}, a_1^{(1)}) \cap V_2(a_2^{(1)}, b_2^{(1)})\right)$$
$$A_0 := \{a^{(1)}\} \cup \left(V_1(a_1^{(1)}, b_1^{(1)}) \cap V_2(a_2^{(1)}, a_2^{(2)})\right)$$
$$B_1 := \{b^{(1)}\} \cup \left(V_1(a_1^{(1)}, \infty) \cap V_2(a_2^{(2)}, b_2^{(1)})\right)$$
$$B_0 := \left(V_1(b_1^{(1)}, \infty) \cap V_2(a_2^{(1)}, a_2^{(2)})\right).$$

If $b_1 > a_1^{(1)}$, define the map $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ 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 5.15. *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. □

Proposition 5.16. *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. □

Proposition 5.17. *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. □

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 5.11, if u, v cover A, then $\mathsf{ch}(G) > \mathsf{ch}(G[A])$.

Proposition 5.18. *If G is connected and $b_1 < a_1^{(1)}$, then for each $D \in \{A_i\}_{i=0}^{\alpha} \cup \{B_i\}_{i=0}^{\beta}$, $\mathsf{ch}(G[D]) < \mathsf{ch}(G)$.*

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 $b^{(2)}$ 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 $1 \le i \le \alpha$, 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 $2 \le i \le \beta$, 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. □

Lemma 5.19. *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 $V(G) = V_1 \cup \cdots \cup V_m$ such that:*

- $\mathsf{ch}(G[V_i]) < \mathsf{ch}(G)$ *for each* $i \in [m]$*, or* $\mathsf{ch}(\overline{G[V_i]}) < \mathsf{ch}(\overline{G})$ *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 5.14. We will let $m = \alpha + \beta + 2$ and let $V_1, \ldots, V_m$ be the sets $\{A_i\}_{i=0}^{\alpha} \cup \{B_i\}_{i=0}^{\beta}$.

Case 1: $b_1 < a_1^{(1)}$. Then $V_1, \ldots, V_m$ is a partition due to Proposition 5.17, and $\mathsf{ch}(G[V_i]) < \mathsf{ch}(G), i \in [m]$ holds by Proposition 5.18. 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 5.12 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 $x_1 \ge a_1^{(j)}$ and $x_2 \le b_2^{(j-1)}$; 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 $V_1, \ldots, V_m$ such that $\mathsf{ch}(\overline{G}[V_i]) < \mathsf{ch}(\overline{G})$ 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. □

Theorem 5.20. *Let $\mathcal{F}$ be a stable subfamily of permutation graphs. Then $\mathcal{F}$ admits a constant-size equality-based labeling scheme, and hence $\mathsf{SK}(\mathcal{F}) = O(1)$.*

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

We apply an argument similar to Lemma 4.5. 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 5.19 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 4.28, 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 $\mathsf{ch}(G') > \mathsf{ch}(G''')$. On the other hand, if G'' is associated with a $\overline{D}$-node and its parent is associated with a D-node, then $\mathsf{ch}(\overline{G'}) > \mathsf{ch}(\overline{G'''})$.

Now consider any P-node associated with G', with child G''. By Lemma 5.19, it holds that either G'' is a bipartite induced subgraph of G' that is a chain graph, or G'' has $\mathsf{ch}(G'') < \mathsf{ch}(G')$ or $\mathsf{ch}(\overline{G''}) < \mathsf{ch}(\overline{G'})$. It is easy to verify that $\mathsf{ch}(\overline{G}) \leq \mathsf{ch}(G) + 1$ 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 $\mathsf{ch}(G''') < \mathsf{ch}(G')$ or $\mathsf{ch}(\overline{G'''}) < \mathsf{ch}(\overline{G'})$ and $\mathsf{ch}(\overline{G}) \leq k+1$, 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 5.13.

2. If G' is a D-node with children $G_1, \dots, G_t$, append the pair $(D \mid i)$ 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 $(\overline{D} \mid i)$ 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 5.19, 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
$$(P, b, \ell_1(x), \ell_2(x), \ell_3(x), \ell_4(x) \mid i, j_1, j_2, j_3, j_4)$$
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
$$(P, b, \ell_1(x), \ell_2(x), \ell_3(x), \ell_4(x) \mid i, j_1, j_2, j_3, j_4)$$
$$(P, b, \ell_1(y), \ell_2(y), \ell_3(y), \ell_4(y) \mid i', j_1', j_2', j_3', j_4'),$$

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 $(D \mid i), (D \mid j)$ or $(\overline{D} \mid i), (\overline{D} \mid j)$. 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 $\ell_t(x), \ell_s(y)$. □

Remark 5.21. 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 Lemma 2.7; this explicit bound would not arise from the alternate proof that goes through the twin-width (proper subfamilies of permutation graphs have bounded twin-width [BKTW20], so we could apply Theorem 1.19).

6 Graph Products

The Cartesian product is defined as follows.

Definition 6.1 (Cartesian Products)**.** Let $d \in \mathbb{N}$ and $G_1, \dots, G_d$ be any graphs. The Cartesian product $G_1 \square G_2 \square \cdots \square G_d$ is the graph whose vertices are the tuples $(v_1, \dots, v_d) \in V(G_1) \times \cdots \times V(G_d)$, 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 $\mathcal{F}$:

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

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.

Theorem 1.17. *If $\mathcal{F}$ is any family that admits a constant-size PUG (including any finite family), then $\mathsf{cl}(\mathcal{F}^\square)$ 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 $\mathsf{cl}(\mathcal{F}^\square)$ is a factorial family with a poly(n)-size universal graph, when $\mathcal{F}$ is any family with a constant-size PUG (including any finite family), which follows by Lemma 2.4 and Proposition 1.7.

Corollary 6.2. *For any hereditary family $\mathcal{F}$ that admits a constant-size PUG (in particular any finite family), the hereditary family $\mathsf{cl}(\mathcal{F}^\square)$ has speed $2^{O(n \log n)}$ and admits a* poly(n)*-size universal graph.*

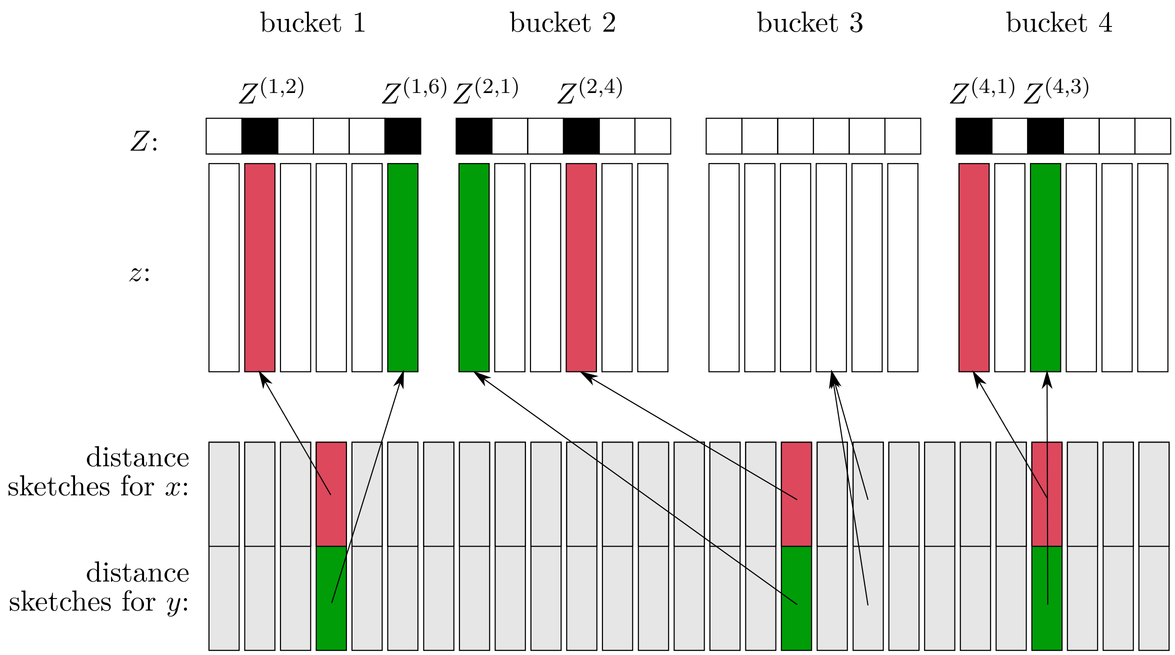

Figure 10: Small-distance sketch for Cartesian products. Along the bottom are the distances sketches for x_i (top) and y_i (bottom) for $i = 1$ to d. Where $x_i = y_i$, the sketches for x_i and y_i are equal and are colored grey; they cancel out in Z and z. Where $x_i \neq y_i$, the sketch for x_i is 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. $c(i, x_i) \neq c(i, y_i)$).

The following proof is illustrated in Figure 10.

Proof of Theorem 1.17. Let $s(k)$ be the size of the randomized distance k labeling scheme for $\mathcal{F}$, and let $D : \{0,1\}^{s(k)} \times \{0,1\}^{s(k)} \to \{0,1\}$ 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 2.3. 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 $\ell_i : V(G_i) \to \{0,1\}^{s(k)}$ 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

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

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 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 $2q \le 2k$ 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 6.3. *Let $x, y \in G$ such that there are exactly $q \le k$ 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 $\mathsf{dist}(x_i, y_i) \le k$ and $\mathsf{dist}(x_i, y_i) = D(\ell_i(x_i), \ell_i(y_i))$, or $\mathsf{dist}(x_i, y_i) > k$ 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 $\mathsf{dist}(x_i, y_i)$ (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 $|I|^2 1/m \le k^2/m \le 1/9$.

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

Claim 6.4. *For any $x, y \in G$ such that there are exactly $q \le k$ 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 6.3 occur, the decoder outputs $\mathsf{dist}(x, y)$ unless there is a coordinate j such that $\mathsf{dist}(x_j, y_j) > k$, 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

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

Multiplying each indicator $\mathbb{1}[(b(j), c(j, x_j)) = (b(i), c(i, x_i))]$ by the vector $\ell_i(x_i)$, and each indicator $\mathbb{1}[(b(j), c(j, y_j)) = (b(i), c(i, x_i))]$ 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

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

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

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 $\mathsf{dist}(x, y)$ 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 6.5 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 $\mathsf{dist}(x_i, y_i) > k$. 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 $\mathsf{dist}(x_i, y_i) \leq k$. Then the above two claims suffices to show that the decoder will output $\mathsf{dist}(x, y)$ with probability at least $2/3$.

This conclues the proof. ◻

Proposition 6.5. *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 ith standard basis vector. Let $R_1, \ldots, R_n \sim [u]$ be uniformly and independently random. Then $\mathbb{P}[|e_{R_1} + \cdots + e_{R_n}| \leq k] < \delta$, 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 $S_1, \ldots, S_t \sim [u]$, we have $\mathbb{P}[|e + e_{S_1} + \cdots + e_{S_t}| \leq k] < \delta$.*

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 $\mathbb{P}[|e + e_{S_1} + \cdots + e_{S_t}| \leq k] = 0 < \delta$, 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 $\mathbb{P}[|e + e_{S_i}| < |e|] \leq \frac{k+t}{u} \leq \frac{2t}{u} \leq \frac{2\sqrt{\delta u}/3}{u} = \frac{2}{3}\frac{\sqrt{\delta}}{\sqrt{u}}$. 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 $|e + e_{S_1} + \cdots + e_{S_t}| = |e| + |e_{S_1}| + \cdots + |e_{S_t}| \geq t > k$, so $\mathbb{P}[|e + e_{S_1} + \cdots + e_{S_t}| \leq k] \leq \mathbb{P}[A \cup B] < \delta$, which proves the claim. ◻

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 $|e_{R_1} + \cdots + e_{R_n}| = n > k$. Hence, $\mathbb{P}[|e_{R_1} + \cdots + e_{R_n}| \leq k] < \delta/2 < \delta$.

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

6.1 Other Graph Products

There are 3 other common types of graph products [HIK11]: the strong product, the direct product, and the lexicographic product, defined as follows.

Definition 6.6 (Other Graph Products)**.** Let G, H be any two graphs, and let $d \in \mathbb{N}$. For each product $\star \in \{\boxtimes, \times, \diamond\}$ defined below, a vertex v in $G \star H$ is a pair $(v_1, v_2) \in V(G) \times V(H)$, and, inductively, a vertex v in the d-wise product $G^{\star d}$ is a tuple $(v_1, \ldots, v_d)$ with each $v_i \in V(G)$. Adjacency is defined as follows.

Strong Product: Two vertices v, w in the strong product $G \boxtimes H$ are adjacent if $v \neq w$ and for each $i \in [2]$ such that $v_i \neq w_i$, it holds that (v_i, w_i) is an edge in the respective graph G or H. By induction, two vertices v, w in $G^{\boxtimes d}$ are adjacent when $v \neq w$ and for each $i \in [d]$ such that $v_i \neq w_i$, it holds that $(v_i, w_i) \in E(G)$.

Direct Product: Two vertices v, w in the direct product $G \times H$ are adjacent when $(v_1, w_1) \in E(G)$ and $(v_2, w_2) \in E(H)$. By induction, two vertices v, w in $G^{\times d}$ are adjacent when $(v_i, w_i) \in E(G)$ for each $i \in [d]$.

Lexicographic Product: Two vertices v, w in the lexicographic product $G \diamond H$ are adjacent when either $(v_1, w_1) \in E(G)$ or $v_1 = w_1$ and $(v_2, w_2) \in E(H)$. By induction, two vertices $v, w \in G^{\diamond d}$ are adjacent when $v \neq w$ and the smallest $i \in [d]$ such that $v_i \neq w_i$ satisfies $(v_i, w_i) \in E(G)$.

For any graph product $\star$ and any base graph G, we define the family $G^\star := \mathsf{cl}\{G^{\star d} : d \in \mathbb{N}\}$.

Unlike the Cartesian products, the families $G^\boxtimes, G^\times, G^\diamond$ may not be stable, depending on the choice of G. We prove the following fact but defer the proof to Appendix B; this boils down to case analysis on the base graph G, and the constant-size PUGs (when possible) are trivial.

Proposition 6.7. *Let $\star \in \{\boxtimes, \times, \diamond\}$ be the strong, direct, or lexicographic product. For any graph G, the family $G^\star$ admits a constant-size PUG if and only if $G^\star$ is stable.*

7 Equality-Based Communication Protocols and Labeling Schemes

An *equality-based communication protocol* is a deterministic protocol that has access to an EQUALITY oracle. Specifically, at each step of the protocol, the players can decide to (exactly) compute an instance of EQUALITY at unit cost. Informally, this means that the nodes of the communication tree are either standard nodes, as in deterministic two-way communication, or EQUALITY nodes. Formally, we define these protocols as follows.

Definition 7.1 (Equality-based Communication Protocol)**.** An *equality-based communication protocol* for a communication problem $f = (f_n)_{n \in \mathbb{N}}$, $f_n : [n] \times [n] \to \{0, 1\}$ is a deterministic protocol of the following form. For each n there is a binary communication tree T_n whose inner nodes are either:

1. *Communication nodes* of the form (p, m), where p is a symbol in $\{A, B\}$ and $m : [n] \to \{0, 1\}$; or,
2. *Equality nodes* of the form (a, b), where $a, b : [n] \to \mathbb{N}$,

and edges are labeled in $\{0, 1\}$. Leaf nodes of the T are labeled with values in $\{0, 1\}$. On input $x, y \in [n]$, the players Alice and Bob perform the following. Each player keeps track of the current node c, which begins at the root of T. The protocol proceeds as follows:

1. If c is a leaf node, the protocol outputs the label of that node.
2. If $c = (p, m)$ is a communication node and $p = A$, then Alice computes $m(x)$ and sends the result to Bob, and both players reset c to be the child labeled with edge value $m(x)$. If $p = B$ then Bob computes $m(y)$ and sends the result to Alice and c becomes the child labeled with edge value $m(y)$.
3. If $c = (a, b)$ is an equality node, then c moves to the child labeled with edge value $\textsc{Eq}[a(x), b(y)]$.

For a communication tree T and inputs x, y, we will write $T(x, y)$ for the output of the protocol T. We will write $\mathsf{CC}^{\text{EQ}}(f_n)$ for the minimum depth of such a tree that computes f_n, and $\mathsf{CC}^{\text{EQ}}(f)$ for the function $n \mapsto \mathsf{CC}^{\text{EQ}}(f_n)$. The equality-based communication protocol is *constant-cost* if $\mathsf{CC}^{\text{EQ}}(f) = O(1)$.

It was observed in [CLV19] that EQUALITY nodes can simulate standard communication nodes. We include a proof for the sake of clarity.

Proposition 7.2. *For any equality-based communication tree T, there is an equality-based communication tree T' with the same depth as T, with all nodes being equality nodes, and such that $T'(x, y) = T(x, y)$ on all inputs x, y.*

Proof. Consider any node (p, m) in T with $p \in \{A, B\}$. If $p = A$, replace this node with an equality node (a, b) where $a(x) = m(x)$ and $b(y) = 1$. If $p = B$, replace this node with an equality node (a, b) where $a(x) = 1$ and $b(y) = m(y)$. We now observe that for any x, y, if $p = A$ then the output of the node is 1 if and only if $1 = m(x)$, and if $p = B$ then the output of the node is 1 if and only if $1 = m(y)$. So the output of this node is the same as the original nodel (p, m). We may replace each node in the tree in such a way to produce T'. □

Recall the definition of equality-based labeling schemes.

Definition 2.5. (Equality-based Labeling Scheme). Let $\mathcal{F}$ be a family 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 $D_{p_1,p_2} : \{0, 1\}^{k \times k} \to \{0, 1\}$ 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 $Q_{x,y}(i, j) = \text{EQ}(q_i(x), q_j(y))$. If $s = 0$ we simply write $D(Q_{x,y})$.

Definition 7.3 (Diagonal Labeling Scheme)**.** We call an equality-based labeling scheme k-*diagonal* if $s = 0$ (so all prefixes are empty, and we write $p(x) = \perp$) and there is a function $\eta : \{0, 1\}^k \to \{0, 1\}$ such that the decoder satisfies $D_{\perp,\perp}(Q_{x,y}) = \eta(Q_{x,y}(1, 1), Q_{x,y}(2, 2), \dots, Q_{x,y}(k, k))$ for all x, y.

Proposition 7.4. *Let $\mathcal{F}$ be any hereditary graph family. If $\mathcal{F}$ admits a constant-size equality-based labeling scheme, then there is a constant-size equality-based protocol for* $\text{ADJ}_{\mathcal{F}}$.

Proof. Suppose there is a constant-size equality-based labeling scheme for $\mathcal{F}$. For any $G \in \mathcal{F}_n$, we can compute the edge relation $g : [n] \times [n] \to \{0, 1\}$ with a protocol as follows. On input x and y, Alice and Bob compute $p(x), p(y)$ and $q_i(x), q_i(y)$ for each $i \in [k]$. Alice sends $p(x)$ to Bob using s bits of communication, and then using k^2 calls to the EQUALITY oracle, they compute each pair $\text{EQ}(q_i(x), q_j(y))$ and construct $Q_{x,y}$. Then Bob outputs $D_{p(x),p(y)}(Q_{x,y})$. □

Lemma 7.5. *Let $\mathcal{F}$ be any hereditary graph family. If there is a constant-cost equality-based communication protocol for* $\text{ADJ}_{\mathcal{F}}$*, then, for some constant k,* $\mathsf{bip}(\mathcal{F})$ *has a k-diagonal labeling scheme.*

Proof. We design a labeling scheme for $\mathsf{bip}(\mathcal{F})$ as follows. Write $d = \mathsf{CC}^{\text{EQ}}(\text{ADJ}_{\mathcal{F}})$, which is a constant. For any $G \in \mathcal{F}_n$, there is an equality-based communication tree T with depth at most d that computes adjacency in G. Write $\mathsf{bip}(G) = (X, Y, E)$ where X, Y are copies of the vertex set of G.

Order the nodes of T such that each vertex precedes its children, and the subtree below the 0-valued edge precedes the subtree below the 1-valued edge. By Proposition 7.2 we may assume that all nodes in T are equality nodes. We may also assume that the tree is complete, and that leaf nodes alternate between 0 and 1 outputs in the order just defined. Let $(a_1, b_1), \dots, (a_t, b_t)$ be the inner nodes of T (which are all equality nodes), in the order just defined.

- For each $x \in X$, define the equality codes $q_i(x) = a_i(x)$ for each $i \in [t]$. Set $q_{t+1}(x) = 0$.
- For each $y \in Y$, define the equality codes $q_i(x) = b_i(x)$ for each $i \in [t]$. Set $q_{t+1}(y) = 1$.

It holds that $t \le 2^d$ since all trees have depth at most d. We define $\eta : \{0,1\}^{t+1} \to \{0,1\}$ as the function that, on input $w \in \{0,1\}^{t+1}$, outputs 0 if $w_{t+1} = 1$, and otherwise simulates the decision tree with node i having output w_i. (Note that we have assumed that all trees are complete, with depth d, with the same outputs on each leaf, so that the output of the tree is determined by the output of each node.) Then on input $x \in X, y \in Y$, we get

$$\begin{aligned}&\eta(\text{EQ}(q_1(x), q_1(y)), \dots, \text{EQ}(q_t(x), q_t(y)), \text{EQ}(0,1))\\ &\quad = \eta(\text{EQ}(a_1(x), b_1(y)), \dots, \text{EQ}(a_t(x), b_t(y))\text{EQ}(0,1)) = T(x,y)\end{aligned}$$

which is 1 if and only if x, y is an edge in $\mathsf{bip}(G)$. On inputs $x, y \in X$ or $x, y \in Y$, we get

$$\eta(\text{EQ}(q_1(x), q_1(y)), \dots, \text{EQ}(q_t(x), q_t(y)), \text{EQ}(0,0)) = 0$$

or

$$\eta(\text{EQ}(q_1(x), q_1(y)), \dots, \text{EQ}(q_t(x), q_t(y)), \text{EQ}(1,1)) = 0$$

respectively, as desired. ☐

7.1 Characterization as Equivalence-Interpretable

In this section we give a characterization of the graph families that admit constant-size equality-based labeling schemes (equivalently, constant-cost equality-based communication protocols). Independently of our work, Hambardzumyan, Hatami, & Hatami [HHH21] gave a different characterization of Boolean matrices (i.e. bipartite graphs) that admit constant-cost equality-based communication protocols: they show that any such matrix M is a linear combination of a constant number of adjacency matrices of bipartite equivalence graphs.

Recall that a graph G is an equivalence graph if it is the disjoint union of cliques, and a colored bipartite graph $G = (X, Y, E)$ is a bipartite equivalence graph if it is a colored disjoint union of bicliques.

Definition 7.6. For a constant $t \in \mathbb{N}$, we say that a family $\mathcal{F}$ of bipartite graphs is *t-equivalence interpretable* if there exists a function $\eta : \{0,1\}^t \to \{0,1\}$, such that the following holds. For every $G = (X, Y, E)$ in $\mathcal{F}$, there exists a vertex-pair coloring $\kappa : X \times Y \to \{0,1\}^t$, where:

1. For every $i \in [t]$, the graph (X, Y, E_i), where $E_i = \{(x, y) \in X \times Y : \kappa(x,y)_i = 1\}$, is a bipartite equivalence graph;
2. For every $x \in X, y \in Y$, $(x, y) \in E$ if and only if $\eta(\kappa(x,y)) = 1$.

Definition 7.7. For a constant $t \in \mathbb{N}$, we say that a family $\mathcal{F}$ of graphs is *strongly t-equivalence interpretable* if there exists a function $\eta : \{0,1\}^t \to \{0,1\}$, such that the following holds. For every $G = (V, E)$ in $\mathcal{F}$, there exists a vertex-pair coloring $\kappa : V \times V \to \{0,1\}^t$, where:

1. $\kappa(x, y) = \kappa(y, x)$ for every pair x, y;
2. For every $i \in [t]$, the graph (V, E_i), where $E_i = \{(x, y) : \kappa(x, y)_i = 1\}$, is an equivalence graph;
3. For every $x, y \in V$, $(x, y) \in E$ if and only if $\eta(\kappa(x, y)) = 1$.

We recall that given a graph $G = (V, E)$, the graph $\mathsf{bip}(G)$ is the bipartite graph (V_1, V_2, E'), where V_1 and V_2 are two copies of V, and for every $x \in V_1$ and $y \in V_2$, $(x, y) \in E'$ if and only if $(x, y) \in E$.

Lemma 7.8. *A hereditary graph family $\mathcal{F}$ has a constant-size equality-based labeling scheme if and only if $\mathsf{bip}(\mathcal{F})$ is t-equivalence interpretable for some constant t.*

Proof. Suppose that $\mathcal{F}$ has a constant-size equality-based labeling scheme. By Proposition 7.4 and Lemma 7.5, $\mathsf{bip}(\mathcal{F})$ has a size t diagonal labeling for some constant t. Let $\eta : \{0,1\}^t \to \{0,1\}$ be the function in the diagonal labeling.

Let $G \in \mathcal{F}$ so that $\mathsf{bip}(G) \in \mathsf{bip}(\mathcal{F})$. Write $\mathsf{bip}(G) = (X, Y, E)$ where X, Y are copies of the vertices of G. Each vertex $x \in X \cup Y$ has a label of the form $(q_1(x), \dots, q_t(x))$.

For each $i \in [t]$ and $x \in X, y \in Y$, define the color $\kappa(x, y)_i = \textsc{Eq}(q_i(x), q_i(y))$. Consider the graph with edges $(x, y) \in X \times Y$ if and only if $\kappa(x, y)_i = 1$. Let $x, x' \in X$ and $y, y' \in Y$ satisfy $\kappa(x, y)_i = \kappa(x', y)_i = \kappa(x', y')_i = 1$, so that (x, y, x', y') forms a path. Then $q_i(x) = q_i(y) = q_i(x') = q_i(y')$, so $\kappa(x, y')_i = 1$ and (x, y') is an edge. So this graph must be P_4-free; i.e. it is a bipartite equivalence graph.

Finally, it holds that for any $x \in X, y \in Y$,

$$\eta(\kappa(x, y)_1, \dots, \kappa(x, y)_t) = \eta(\textsc{Eq}(q_1(x), q_1(y)), \dots, \textsc{Eq}(q_t(x), q_t(y)))$$

which is 1 if and only if x, y is an edge in $\mathsf{bip}(G)$.

Now suppose that $\mathsf{bip}(\mathcal{F})$ is t-equivalence interpretable. It suffices to construct a labeling scheme for the graphs $\mathsf{bip}(G)$ for $G \in \mathcal{F}$. Write $\mathsf{bip}(G) = (X, Y, E)$ and let $\kappa(x, y) \in \{0,1\}^t$ be the coloring of vertex pairs $x \in X, y \in Y$. For each $i \in [t]$ we let E_i be the edge set of the equivalence graph such that $(x, y) \in E_i$ if and only if $\kappa(x, y)_i = 1$. Give an arbitrary numbering to the bicliques in E_i and define $q_i(x)$ to be the number of the biclique to which x belongs. It then holds that for any $x \in X, y \in Y$,

$$\eta(\textsc{Eq}(q_1(x), q_1(y)), \dots, \textsc{Eq}(q_t(x), q_t(y))) = \eta(\kappa(x, y)_1, \dots, \kappa(x, y)_t)\,,$$

which is 1 if and only if (x, y) is an edge of $\mathsf{bip}(G)$. Therefore we have obtained a t-diagonal labeling scheme. □

Proposition 7.9. *Let $\mathcal{F}$ be a hereditary family of (uncolored) bipartite graphs. If $\mathsf{bip}(\mathcal{F})$ is t-equivalence interpretable then $\mathcal{F}$ is strongly $(t+1)$-equivalence interpretable.*

Proof. Let $\eta : \{0,1\}^t \to \{0,1\}$ be the function that witnesses $\mathcal{F}$ as t-equivalence interpretable and consider any bipartite graph $G \in \mathcal{F}$ with a fixed bipartition (X, Y) of its vertices. Then $\mathsf{bip}(G) = (X_1 \cup Y_1, X_2 \cup Y_2, E' \cup E'')$ is the disjoint union of $G' = (X_1, Y_2, E')$ and $G'' = (X_2, Y_1, E'')$, where G' and G'' are each isomorphic to G. By definition of equivalence-interpretable, there are graphs $B_1, \ldots, B_t$ with parts $X_1 \cup Y_1$ and $X_2 \cup Y_2$ where each B_i is a bipartite equivalence graph, and such that each pair $x \in X_1 \cup Y_1, y \in X_2 \cup Y_2$ is an edge of $\mathsf{bip}(G)$ if and only if

$$\eta(B_1(x,y), \ldots, B_t(x,y)) = 1,$$

where $B_i(x,y) = 1$ if (x,y) is an edge in B_i, and $B_i(x,y) = 0$ otherwise. Clearly, each B_i induces a bipartite equivalence graph when restricted the vertices of G'. We now consider graphs B_i' on vertex set $X_1 \cup Y_2$ where (x,y) is an edge if and only if x, y belong to the same biclique in B_i. B_i' is an equivalence graph since it is obtained by taking a disjoint union of bicliques and connecting every two vertices belonging to one of the bicliques. Now define the graph B_{t+1}' on vertices $X_1 \cup Y_2$ such that (x,y) is an edge if and only if $x, y \in X_1$ or $x, y \in Y_2$, so that B_{t+1}' is an equivalence graph. We define the function $\eta' : \{0,1\}^{t+1} \to \{0,1\}$ by setting $\eta'(w) = 0$ if $w_{t+1} = 1$ and otherwise setting $\eta'(w) = \eta(w_1, \ldots, w_t)$. It then holds that for every $(x,y) \in X_1 \times Y_2$,

$$\eta'(B_1'(x,y), \ldots, B_t'(x,y), B_{t+1}'(x,y)) = \eta(B_1(x,y), \ldots, B_t(x,y))$$

which is 1 if and only if x, y are adjacent in G'. On the other hand, for $x, y \in X_1$ or $x, y \in Y_2$, we have $\eta'(B_1'(x,y), \ldots, B_t'(x,y), B_{t+1}'(x,y)) = 0$, so the same property holds. Consequently, G' is strongly $(t+1)$-equivalence interpretable. Since G is isomorphic to G', it is also strongly $(t+1)$-equivalence interpretable. □

7.2 Cartesian Products Do Not Admit an Equality-Based Labeling Scheme

We now prove the following theorem from the introduction. This will follow from Theorem 7.12, which proves the same statement for equality-based labeling schemes. This is equivalent, by Proposition 7.4.

Theorem 1.18. *There is no constant-cost equality-based protocol for computing adjacency in P_2^n.*

The d-dimensional hypercube H_d is the d-wise Cartesian product $P_2^{\square d}$ of the single edge. The family $\mathcal{H} = \mathsf{cl}(\{H_d : d \in \mathbb{N}\})$ of induced subgraphs of the hypercubes is sometimes called the family of *cubical* graphs. In this section, we will use Lemma 7.8 to prove that the class of cubical graphs does not admit an equality-based labeling scheme. In our proof, we will employ some results from the literature. We denote by C_n the n-vertex cycle.

Theorem 7.10 ([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* $\mathsf{eqc}(G)$ 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.

Theorem 7.11 ([LNP80, Alo86])**.** *For every $n \geq 3$, it holds that $\mathsf{eqc}(\overline{C_n}) \geq \log n - 1$.*

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 also write $x \prec y$ if $x \preceq y$ and $x \neq y$.

Theorem 7.12. *The class $\mathcal{H}$ does not admit a constant-size equality-based labeling scheme.*

Proof. Suppose, towards a contradiction, that $\mathcal{H}$ admits a constant-size equality-based labeling scheme. Then, since $\mathcal{H}$ is a family of bipartite graphs, by Lemma 7.8 and Proposition 7.9, there exists a t such that $\mathcal{H}$ is strongly t-equivalence interpretable.

Let $k = 2^t$, $\ell = 2^{t+1}$ and let $n \geq n_0(k,\ell)$, where $n_0(k,\ell)$ is the function from Theorem 7.10. Let V and E be the vertex and the edge sets of the hypercube H_n respectively. Let $\kappa : V \times V \to \{0,1\}^t$

and $\eta : \{0,1\}^t \to \{0,1\}$ be the functions as in Definition 7.7 witnessing that the hypercube H_n is strongly t-equivalence interpretable. Color every edge (a,b) of H_n with $\kappa(a,b)$. Since the edges of H_n are colored in at most $k = 2^t$ different colors, by Theorem 7.10 it contains a monochromatic induced cycle $C = (V', E')$ of length $2\ell = 2^{t+2}$. Let $\kappa^* \in \{0,1\}^t$ be the color of the edges of C.

Claim 1. *For every distinct $a, b \in V'$ that are* not *adjacent in C, we have $\kappa^* \prec \kappa(a,b)$.*

Proof. Since every connected component of an equivalence graph is a clique, it follows that for every $i \in [t]$, $\kappa^*_i = 1$ implies $\kappa(x,y)_i = 1$ for every $x, y \in V'$. Hence, $\kappa^* \preceq \kappa(a,b)$. Furthermore, since a and b are not adjacent in C, we have that $\kappa(a,b) \neq \kappa^*$, as otherwise we would have $\eta(\kappa(a,b)) = \eta(\kappa^*) = 1$ and hence a and b would be adjacent. □

Let now $I \subseteq [t]$ be the index set such that $i \in I$ if and only if $\kappa^*_i = 0$ and there exist $a, b \in V'$ with $\kappa(a,b)_i = 1$. For every $i \in I$ let $F_i = (V', E'_i)$, where $E'_i = \{(a,b) \mid a, b \in V', \kappa(a,b)_i = 1\}$. Clearly, all these graphs are equivalence graphs. 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 of C, in other words the union coincides with $\overline{C}$. Thus $\mathsf{eqc}(\overline{C}) \leq |I| \leq t$. However, by Theorem 7.11, $\mathsf{eqc}(\overline{C}) \geq \log |V'| - 1 \geq t + 1$, a contradiction. □

8 Twin-width

In this section, we prove the following. Twin-width is formally defined in Definition 8.1.

Theorem 1.19. *Let $\mathcal{F}$ be a hereditary family of graphs with bounded twin-width. Then $\mathcal{F}$ admits a constant-size PUG if and only if $\mathcal{F}$ is stable.*

To prove the theorem, we will first reduce the problem to bipartite graphs in Section 8.2, and then show in Section 8.3 how to construct a constant-size equality-based labeling scheme for any stable family of bipartite graphs of bounded twin-width. In Section 8.4 we will generalize the above theorem to first-order labeling schemes. As a consequence we will obtain an answer to an open problem from [Har20] about the existence of constant-size distance sketches for planar graphs. In Section 8.1 we provide necessary definitions and notations.

8.1 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 8.1 (Twin-Width)**.** An *uncontraction sequence of width d* of a graph $G = (V, E)$ is a sequence $\mathcal{P}_1, \ldots, \mathcal{P}_m$ of partitions of V such that:

- $\mathcal{P}_1 = \{V\}$;
- $\mathcal{P}_m$ is a partition into singletons;
- For $i = 1, \ldots, m-1$, $\mathcal{P}_{i+1}$ 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* $\mathsf{tww}(G)$ 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. [GPT21]. 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 family $\mathcal{F}$ is the family of all ϕ-transductions of graphs in $\mathcal{F}$. A graph family $\mathcal{G}$ is an FO-transduction of $\mathcal{F}$ if there exists an FO formula ϕ so that $\mathcal{G}$ is the ϕ-transduction of $\mathcal{F}$.

Theorem 8.2 ([BKTW20, NMP+21]). *Let $\mathcal{F}$ be a stable family of bounded twin-width. Then any FO transduction of $\mathcal{F}$ is also a stable family of bounded twin-width.*

8.2 From General Graphs to Bipartite Graphs

Proposition 8.3. *Let $\mathcal{F}$ be a family of graphs. Then the family $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ is an FO transduction of $\mathcal{F}$.*

Sketch of the proof. Let $G = (V, E)$ be a graph in $\mathcal{F}$. To obtain the graph $\mathsf{bip}(G)$, 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 $\mathsf{bip}(G)$, we in addition take an induced subgraph. □

We can argue now that to prove Theorem 1.19 it is enough to establish its special case for the bipartite families. Indeed, assume that Theorem 1.19 holds for the bipartite families and let $\mathcal{F}$ be an arbitrary hereditary stable family of graphs of bounded twin-width. Then, by Proposition 8.3 and Theorem 8.2, the family $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ is also stable and has bounded twin-width. Therefore, by Proposition 4.3 (4), any constant-size equality-based labeling scheme for $\mathsf{cl}(\mathsf{bip}(\mathcal{F}))$ can be turned into a constant-size equality-based labeling scheme for $\mathcal{F}$.

8.3 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 8.4 (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 8.5 (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 8.6 (Convex Twin-Width)**.** A *convex uncontraction sequence of width* d of an ordered bipartite graph $G = (X, Y, E, \leq)$ is a sequence $\mathcal{P}_1, \ldots, \mathcal{P}_m$ of divisions of $X \cup Y$ such that:

- $\mathcal{P}_1 = \{X, Y\}$;
- $\mathcal{P}_m$ is a division into singletons;
- For $i = 1, \ldots, m-1$, 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* $\mathsf{ctww}(G)$ is the minimum d such that there is a convex uncontraction sequence of width d of G.

Lemma 8.7 ([GPT21], Lemmas 3.14 & 3.15)**.** *For any ordered bipartite graph* $G, \mathsf{tww}(G) \leq \mathsf{ctww}(G)$*. For any bipartite graph* $G = (X, Y, E)$*, there is a total order* $\leq$ *on* $X \cup Y$ *such that* $\mathsf{ctww}((X, Y, E, \leq)) \leq \mathsf{tww}(G) + 1$*.*

Definition 8.8 (Quasi-Chain Number)**.** Let $G = (X, Y, E)$ be a bipartite graph. The *quasi-chain number* $\mathsf{qch}(G)$ of G is the largest k for which there exist two sequences $x_1, \ldots, x_k \in X$ and $y_1, \ldots, y_k \in Y$ 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 $y_1, \ldots, y_{i-1}$ and y_i is non-adjacent to all of $x_1, \ldots, x_{i-1}$; or,
2. x_i is non-adjacent to all of $y_1, \ldots, y_{i-1}$ and y_i is adjacent to all of $x_1, \ldots, x_{i-1}$.

Lemma 8.9 ([GPT21], Lemma 3.3)**.** *For every bipartite graph* G*,*

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

Definition 8.10 (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 $G = G_0, G_1, \ldots, G_r = G'$ such that $r \leq q$ and for each $i \in [r]$, G_i is a flip of G_{i-1}.

Lemma 8.11 ([GPT21], 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* $\mathcal{U}_1^X, \ldots, \mathcal{U}_r^X \subseteq \mathcal{D}^X$*,* $\mathcal{U}_1^Y, \ldots, \mathcal{U}_r^Y \subseteq \mathcal{D}^Y$*, 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* $K_1, \ldots, K_m$*, the quasi-chain number of* $G[C \cup K_1 \cup \cdots \cup K_m]$ *is at most* $k - 1$*.*

Lemma 8.12. *Let* $\mathcal{F}$ *be any family 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 $\mathsf{ch}(\mathcal{F}) = k$ and $\mathsf{tww}(\mathcal{F}) = d$ for some constants k, d.

Decomposition tree. For any $G \in \mathcal{F}$, we construct a decomposition tree as follows. Let $k^* = \mathsf{qch}(G)$ which satisfies $k^* \le 4 \cdot \mathsf{qch}(G) + 4$ by Lemma 8.9. 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 $\mathsf{qch}(G') \le 1$, we have $\mathsf{ch}(G') \le \mathsf{qch}(G') \le 1$ by Lemma 8.9, so G' is P_4-free.

For a node $G' = (X', Y', E')$ with $\mathsf{qch}(G') = k' > 1$, let $G'_{\le} = (X', Y', E', \le)$ be an ordered bipartite graph with $\mathsf{ctww}(G'_{\le}) \le d + 1$, which exists due to Lemma 8.7. Let $\mathcal{D}$ be a division of $G'_{\le}$, 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 $G'_{\le}$ 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 $\mathsf{qch}(G'[C \cup K_1 \cup \dots \cup K_m]) \le k' - 1$.

The above holds for some r, q determined by Lemma 8.11. 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 $\mathsf{qch}(G'') < \mathsf{qch}(G')$, so the depth of the tree is at most $\mathsf{qch}(G) = k^* \le 4k + 4$.

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 $\mathsf{qch}(G') \le 1$, 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 $G'_{\le} = F_0, F_1, \dots, F_{q'} = F$ with $q' \le q$ be the sequence of flips that take $G'_{\le}$ 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 $G'_i = G'[C \cup K_1 \cup \ldots \cup K_m]$ 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 $L(x, G') = (0, p(x) \mid q(x))$, $L(y, G') = (0, p(y) \mid q(y))$ then output the adjacency of x, y in G', which is a bipartite equivalence graph, as determined by the labels $(p(x) \mid q(x))$, $(p(y) \mid q(y))$.

2. If $L(x,G') = (1, f(x) \mid s_1(S_1), \ldots, s_r(S_r))$, $L(y,G') = (1, f(y) \mid s_1(S'_1), \ldots, s_r(S'_r))$ 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
$$\left|\{i \in [q] : f(x)_i = f(y)_i = 1\}\right|.$$

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

Claim 8.13. *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 $\{A, K_1, \ldots, K_m\} \subseteq \mathcal{U}_i$ such that $B \in \{K_1, \ldots, K_m\}$. 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 $G'' = G'[C \cup K_1 \cup \cdots \cup K_m]$ that contains both x and y. □

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

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 $G_0, \ldots, G_t$ 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
$$\left|\{i \in [q] : f(x)_i = f(y)_i = 1\}\right|.$$
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 $(G_t)_{\leq} = F_0, F_1, \ldots, F_{q'} = F$; 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 $i = 0, \ldots, t-1$. By definition, there is a child G_{i+1} that contains both x and y, so there exists $j \in [r]$ and a star $S = (C, K_1, \ldots, K_m)$ in $H[\mathcal{U}_j^X, \mathcal{U}_j^Y]$ such that $x, y \in C \cup K_1 \cup \cdots \cup K_m$. Then $s_j(S_j) = s_j(S'_j) = s_j(S)$ so the decoder will recurse on the child G_{i+1}. □

8.4 First-Order Labeling Schemes & Distance Sketching

In this section we construct sketches for stable graph families 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 $\operatorname{arity}(R) \geq 1$. 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 family $\mathcal{F}$, we will write $\mathcal{F}^\tau$ 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 8.14 (First-order sketches)**.** For a symmetric first-order formula $\phi(x, y)$ with vocabulary τ, and a graph family $\mathcal{F}$, 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 $\mathsf{sk} : V \to \{0,1\}^{c(n)}$. The encoder and (deterministic) decoder $D : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ must satisfy the condition that for all $G \in \mathcal{F}^\tau$,

$$\forall u, v \in V(G) : \qquad \mathbb{P}_{\mathsf{sk}}\Big[D\big(\mathsf{sk}(u), \mathsf{sk}(v)\big) = \mathbb{1}[G \models \phi[u/x, v/y]]\Big] \;\geq\; 1 - \delta\,.$$

As usual, if left unspecified, we assume $\delta = 1/3$. We will write ϕ-$\mathsf{SK}(\mathcal{F})$ 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 8.15. *Let $\mathcal{F}$ be a stable, hereditary graph family with bounded twin-width, and let $\phi(x, y)$ be an FO formula of vocabulary τ. Then ϕ-$\mathsf{SK}(\mathcal{F}) = O(1)$.*

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 $\mathcal{F}^\phi := \mathsf{cl}(\{G^\phi : G \in \mathcal{F}^\tau\})$. By definition, $\mathcal{F}^\phi$ is hereditary. Furthermore, we note that $\mathcal{F}^\phi$ is an FO transduction of $\mathcal{F}$. Hence, Theorem 8.2 implies that $\mathcal{F}^\phi$ is a stable family of bounded twin-width, and therefore by Theorem 1.19, $\mathcal{F}^\phi$ 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. □

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

Corollary 8.16. *For any $k \in \mathbb{N}$ there is a constant-size sketch for planar graphs that decides* $\mathsf{dist}(x,y) \le k$*.*

Proof. The family of planar graphs has bounded twin-width [BKTW20] and bounded chain number (because it is of bounded degeneracy). Hence, the result follows from Theorem 8.15 and the fact that the relation $\mathsf{dist}(x,y) \le k$ is expressible in first-order logic, e.g. via the following FO formula

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

$\square$

A Proofs from Section 3

In this section we prove the following theorem from Section 3.

Theorem 3.6. *Let $\mathcal{F}$ be a hereditary graph family. Then:*

1. *If $\mathcal{F}$ is a minimal family 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).*

A.1 Tools

We will use the following tools to prove our results for the families below the Bell numbers, and the minimal families above the Bell numbers.

Proposition A.1 (Bounded vertex addition)**.** *Let $c \in \mathbb{N}$ and let $\mathcal{X}$ be a family of graphs. Denote by $\mathcal{F}$ the family of all graphs each of which can be obtained from a graph in $\mathcal{X}$ by adding at most c vertices. If $\mathcal{X}$ admits a constant-size equality-based labeling scheme, then so does $\mathcal{F}$.*

Proof. Let $G \in \mathcal{F}$ and let $W \subseteq V(G)$ be such that $G[V \setminus W] \in \mathcal{X}$ and $|W| \le c$. Given a constant-size adjacency sketch for $G[V \setminus W]$ we will construct a constant-size adjacency sketch for G. Identify the vertices W with numbers $[c]$. For every vertex $x \in V \setminus W$, assign the label $(0, a(x), p(x) \mid q(x))$ where $a(x) \in \{0,1\}^c$ satisfies $a(x)_i = 1$ if and only if x is adjacent to vertex $i \in [c]$, and $(p(x) \mid q(x))$ is the label of x in $G[V \setminus W]$. For every vertex $i \in W = [c]$, assign the label $(1, i, a(i) \mid -)$. On inputs $(0, a(x), p(x) \mid q(x))$ and $(0, a(y), p(y) \mid q(y))$ for $x, y \in V \setminus W$, the decoder for $\mathcal{F}$ simulates the decoder for $\mathcal{X}$ on $(p(x) \mid q(x))$ and $(p(y) \mid q(y))$. On inputs $(0, a(x), p(x) \mid q(x))$ and $(1, i, a(i) \mid -)$, the decoder outputs $a(x)_i$. On inputs $(1, i, a(i) \mid -)$ and $(1, j, a(j) \mid -)$ the decoder outputs $a(i)_j$. $\square$

Definition A.2. Let $k \in \mathbb{N}$ and $\mathcal{F}$ a graph family. We denote by $\mathcal{S}(\mathcal{F}, k)$ the family of all graphs that can be obtained by choosing a graph $G \in \mathcal{F}$, partitioning $V(G)$ into at most k sets $V_1, V_2, \dots, V_r$, $r \le k$, and complementing edges between some pairs of sets V_i, V_j, $i \neq j$, and within some of the sets V_i.

Proposition A.3 (Bounded complementations)**.** *Let $k \in \mathbb{N}$ and let $\mathcal{F}$ be a family that admits a constant-size equality-based labeling scheme. Then $\mathcal{S}(\mathcal{F}, k)$ admits a constant-size equality-based labeling scheme.*

Proof. Let G be a graph in $\mathcal{S}(\mathcal{F}, k)$ that is obtained from a graph in $H \in \mathcal{F}$ by partitioning $V(H)$ into at most k subsets and complementing the edges between some pairs of sets and also within some of the sets. To construct a constant-size equality-based labeling for G, we use a constant-size equality-based labeling for H and extend the label of every vertex by an extra $\lceil \log k \rceil + k$ bits. The first $\lceil \log k \rceil$ of these extra bits are used to store the index of the subset in the partition to which the vertex belongs, and the remaining k bits are used to store the the information of whether the edges within the vertex's partition class are complemented or not and also whether the edges between the vertex's partition class and each of the other partition classes are complemented or not.

Now given new labels of two vertices we first extract the old labels and apply the decoder to infer the adjacency in H. Then we use the extra information about the partition classes and their complementations to deduce whether the adjacency needs to be flipped or not. ☐

A.2 Classes Below the Bell Number

We need the following definition for describing the structure of hereditary families below the Bell number.

Definition A.4. Let k be a positive integer, let D be a graph with loops allowed on the vertex set $[k]$, and let F be a simple graph on the same vertex set $[k]$. Let H' be the disjoint union of infinitely many copies of F, and for $i = 1, \dots, k$, let V_i be the subset of $V(H')$ containing vertex i from each copy of F. Now we define H to be the graph obtained from H' by connecting two vertices $u \in V_i$ and $v \in V_j$ if and only if (u, v) is an edge in H' but not an edge in D, or (u, v) is not an edge in H' but an edge in D. Finally, we denote by $\mathcal{R}(D, F)$ the hereditary class consisting of all the finite induced subgraphs of H.

To better explain the above definition, we note that the infinite graph H' consists of k independent sets $V_1, V_2, \dots, V_k$ such that a pair of distinct sets V_i, V_j induce a perfect matching if (i, j) is an edge in F, and V_i, V_j induce a graph without edges otherwise. The connected components of H' are each isomorphic to F, so H' has maximum degree at most k. Then the graph H is obtained from H' by complementing $H'[V_i]$ whenever i has a loop in D, and applying the bipartite complementation to $H'[V_i, V_j]$ whenever (i, j) is an edge in D.

For any $k \in \mathbb{N}$, let $\mathcal{R}(k) = \bigcup \mathcal{R}(D, F)$, where the union is over all graphs D, F satisfying Definition A.4. We show the following result.

Proposition A.5. *For any natural number k, the family $\mathcal{R}(k)$ admits a constant-size equality-based labeling scheme.*

Proof. From the description above, it follows that H, and therefore any of its induced subgraphs, can be partitioned into at most k sets, each of which is either a clique or an independent set, and the bipartite graph spanned by the edges between any pair of sets is either of maximum degree at most 1 or of maximum co-degree at most 1, i.e. the complement has degree at most 1. Observe that by applying to H (respectively, any of its induced subgraphs) the same complementations according to D again, we turn the graph into H' (respectively an induced subgraph of H'), i.e. to a graph of degree at most k. This shows that, for $\mathcal{Y}_k$ the family of graphs with maximum degree at most k, $\mathcal{R}(k) \subseteq \mathcal{S}(\mathcal{Y}_k, k)$. The claim then follows from Lemma 2.13 (with Remark 2.14) and Proposition A.3. ☐

Lemma A.6 ([BBW00, BBW05])**.** *For every hereditary class $\mathcal{F}$ below the Bell numbers, there exist constants c, k such that for all $G \in \mathcal{F}$ there exists a set W of at most c vertices so that $G[V \setminus W]$ belongs to $\mathcal{R}(D, F)$ for some k-vertex graphs D and F.*

Corollary A.7. *Any hereditary family $\mathcal{F}$ below the Bell numbers admits a constant-size equality-based labeling scheme. Therefore* $\mathsf{SK}(\mathcal{F}) = O(1)$.

Proof. Let $\mathcal{F}$ be a hereditary family below the Bell numbers, and let c and k be natural numbers as in Lemma A.6, i.e. for every graph G in $\mathcal{F}$ there exist a k-vertex graph D with loops allowed and a simple k-vertex graph F so that after removing at most c vertices from G we obtain a graph from $\mathcal{R}(D, F) \subseteq \mathcal{R}(k)$.

By Proposition A.1, $\mathcal{F}$ admits a constant-size equality-based labeling scheme if $\mathcal{R}(k)$ does, and the latter follows from Proposition A.5. □

A.3 Minimal Classes Above the Bell Number

We denote by $\mathcal{P}$ the class of path forests, i.e. graphs in which every component is a path. The following theorem of [BBW05, ACFL16] enumerates the minimal hereditary families above the Bell number.

Theorem A.8 ([BBW05, ACFL16])**.** *Let $\mathcal{F}$ be a minimal hereditary family above the Bell numbers, i.e. every proper hereditary subfamily of $\mathcal{F}$ is below the Bell numbers. Then either $\mathcal{F} \subseteq \mathcal{S}(\mathcal{P}, k)$ for some integer k, or $\mathcal{F}$ is one of the following 13 families:*

(1) *The family $\mathcal{K}_1$ of all graphs whose connected components are cliques (equivalence graphs);*

(2) *The family $\mathcal{K}_2$ of all graphs whose connected components are stars (star forests);*

(3) *The family $\mathcal{K}_3$ of all graphs whose vertices can be partitioned into an independent set I and a clique Q, such that every vertex in Q has at most one neighbor in I;*

(4) *The family $\mathcal{K}_4$ of all graphs whose vertices can be partitioned into an independent set I and a clique Q, such that every vertex in I has at most one neighbor in Q;*

(5) *The family $\mathcal{K}_5$ of all graphs whose vertices can be partitioned into two cliques Q_1, Q_2, such that every vertex in Q_2 has at most one neighbor in Q_1*

(6) *The families $\overline{\mathcal{K}_i}$ for $i \in [5]$, where $\overline{\mathcal{K}_i}$ is the family of complements of graphs in $\mathcal{K}_i$;*

(7) $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}$, *or* $\mathcal{C}^{\bullet\bullet}$.

Corollary A.9. *Any minimal hereditary family $\mathcal{F}$ above the Bell numbers, except $\mathcal{C}^{\circ\circ}, \mathcal{C}^{\bullet\circ}$, and $\mathcal{C}^{\bullet\bullet}$, admits a constant-size equality-based labeling scheme.*

Proof. If $\mathcal{F} \subseteq \mathcal{S}(\mathcal{P}, k)$ for some k, then the result follows from Proposition A.3 and Lemma 2.13 as the graphs in $\mathcal{P}$ have arboricity 1. The result for $\mathcal{K}_1$ follows by Fact 2.11. The result for $\mathcal{K}_2$ follows from Lemma 2.13 as the graphs in $\mathcal{K}_2$ have arboricity 1. The result for $\mathcal{K}_3$, $\mathcal{K}_4$, and $\mathcal{K}_5$ follows from Proposition A.3 as each of these classes is a subclass of $\mathcal{S}(\mathcal{K}_2, 2)$. The result for $\overline{\mathcal{K}_i}$, $i \in [5]$ also follows from Proposition A.3 as $\overline{\mathcal{K}_i} \subseteq \mathcal{S}(\mathcal{K}_i, 1)$ for every $i \in [5]$. □

B Proofs from Section 6

In this section we prove the following from Section 6.

Proposition 6.7. *Let $\star \in \{\boxtimes, \times, \diamond\}$ be the strong, direct, or lexicographic product. For any graph G, the family $G^\star$ admits a constant-size PUG if and only if $G^\star$ is stable.*

Throughout the section we use the fact that for any G, any induced subgraph H of G, any $d \in \mathbb{N}$, and $\star \in \{\square, \boxtimes, \times, \diamond\}$, the graph $H^{\star d}$ is an induced subgraph of $G^{\star d}$, and hence $\mathsf{cl}(H^\star) \subseteq \mathsf{cl}(G^\star)$.

B.1 Strong Products

Proposition B.1. *For any graph G, either $P_3^{\boxtimes} \subseteq G^{\boxtimes}$ or G is an equivalence graph.*

Lemma B.2.

(1) *If G is an equivalence graph, then $\mathsf{cl}(G^{\boxtimes})$ is an equivalence graph.*

(2) *$\mathsf{cl}(P_3^{\boxtimes})$ is the family of all graphs.*

Proof of statement 1. Suppose for contradiction that vertices x, y, z induce P_3 in $H \in \mathsf{cl}(G^{\boxtimes})$, so x, y, z induce P_3 in some $G^{\boxtimes d}$. Assume that $x, y \in V(G^{\boxtimes d})$ are adjacent and $y, z \in V(G^{\boxtimes d})$ are adjacent, and consider any $i \in [d]$. Since x, y are adjacent, it must be that $x_i = y_i$ or x_i, y_i belong to the same connected component of G. Since y, z are adjacent, it must be that $y_i = z_i$ or y_i, z_i belong to the same connected component of G. But then $x_i = z_i$ or x_i, z_i belong to the same connected component of G. This holds for all $i \in [d]$, so x, z are adjacent, a contradiction. So H is P_3-free, which implies that it is an equivalence graph. □

Proof of statement 2. Identify the vertices of P_3 with the numbers $[3]$ so that $(1,2), (2,3)$ are the edges of P_3. Let G be any graph with n vertices. We will show that G is an induced subgraph of $P_3^{\boxtimes d}$ for $d = \binom{n}{2}$. Identify every coordinate $i \in [d]$ with an unordered pair $\{a, b\}$ of vertices in G and write $i(a, b)$ for the coordinate identified with this pair.

We map a vertex $v \in V(G)$ to a vertex x in $P_3^{\boxtimes d}$ as follows. For every $w \in V(G)$, if $(v, w) \in E(G)$ or $v < w$ assign $x_{i(v,w)} = 1$, otherwise, i.e. if $(v, w) \notin E(G)$ and $v > w$ assign $x_{i(v,w)} = 3$. For every pair $a, b \in V(G)$ such that $v \notin \{a, b\}$ assign $x_{i(a,b)} = 2$.

Let $u, v \in V(G)$ be distinct vertices and let x, y be the vertices in $P_3^{\boxtimes d}$ obtained from u, v by the above construction. For every a, b such that either $u \notin \{a, b\}$ or $v \notin \{a, b\}$ it holds that either $x_{i(a,b)} = 2$ or $y_{i(a,b)} = 2$, and in both cases we have either $x_{i(a,b)} = y_{i(a,b)} = 2$ or $(x_{i(a,b)}, y_{i(a,b)}) \in E(P_3)$. Only the case $\{a, b\} = \{u, v\}$ remains, in which $(u, v) \in E(G)$ if and only only if $x_{i(a,b)} = y_{i(a,b)} = 1$, and otherwise $\{x_{i(a,b)}, y_{i(a,b)}\} = \{1, 3\}$ so $x_{i(a,b)}, y_{i(a,b)}$ are not adjacent in P_3. Therefore, u, v are adjacent if and only if x, y are adjacent. □

B.2 Direct Product

We consider the following cases.

Proposition B.3. *For every G, one of the following holds:*

(1) *G contains K_3 or P_4 as an induced subgraph, and therefore $\mathsf{cl}(G^{\times})$ contains at least one of $\mathsf{cl}(K_3^{\times})$ or $\mathsf{cl}(P_4^{\times})$; or*

(2) *G is a bipartite equivalence graph.*

Proof. Assume that G is (K_3, P_4)-free. It follows that G excludes all cycles except C_4, which implies that G is a P_4-free bipartite graph, which is a bipartite equivalence graph. □

The following lemma is sufficient to establish Conjecture 1.2 for all direct product completion families $\mathsf{cl}(G^{\times})$.

Lemma B.4.

(1) *$\mathsf{cl}(K_3^{\times})$ contains all bipartite graphs.*

(2) *$\mathsf{cl}(P_4^{\times})$ contains all bipartite graphs.*

(3) *If G is a bipartite equivalence graph then $\mathsf{cl}(G^{\times})$ is a family of bipartite equivalence graphs.*

Proof of statement 1. Let $G = (X, Y, E)$ be a bipartite graph with $|X| = |Y| = n$. We will show that G is an induced subgraph of $K_3^{\times(n+1)}$. Identify the vertices of K_3 with $[3]$.

Write $X = \{x_1, \ldots, x_n\}, Y = \{y_1, \cdots, y_n\}$. Map each x_i to the vertex $v = (v_0, v_1, \ldots, v_n)$ where $v_0 = 1$, and for each $j \geq 1$ set $v_j = 1$ if $y_j \in N(x_i)$ and $v_j = 2$ if $y_j \notin N(x_i)$. Map each y_i to vertex $w = (w_0, w_1, \ldots, w_n)$ where $w_0 = 2$, $w_i = 2$, and for all $j \notin \{0, i\}$, $w_j = 3$.

Let $p, q \in [n]$ and v, w be the images of x_p and y_q under the above mapping respectively. Observe that v, w are adjacent unless there is some coordinate $j \geq 1$ such that $v_j = w_j$. Since $v_j \in \{1, 2\}$ and $w_j \in \{2, 3\}$, v, w are adjacent unless $v_j = w_j = 2$ for some $j \geq 1$. The latter occurs if and only if $j = q$ and $y_j \neq N(x_p)$, i.e. when x_p and y_q are not adjacent.

Finally, note that the images of any $x_i, x_j \in X$ or any $y_i, y_j \in Y$ are equal in coordinate 0, and so they are not adjacent in $K_3^{\times(n+1)}$. □

Proof of statement 2. Let again $G = (X, Y, E)$ be a bipartite graph with $|X| = |Y| = n$. We will show that G is an induced subgraph of $P_4^{\times(n+1)}$. Identify the vertices of P_4 with the numbers $[4]$, with edges $(1, 2), (2, 3), (3, 4)$.

Let $G = (X, Y, E)$ be any bipartite graph with $|X| = |Y| = n$ vertices and write $X = \{x_1, \ldots, x_n\}, Y = \{y_1, \ldots, y_n\}$. For each vertex $x_i \in X$, let $N(x_i) \subseteq Y$ be its set of neighbors. Map x_i to $v = (v_0, v_1, \ldots, v_n)$ as follows. Set $v_0 = 1$. For each $y_j \in N(x_i)$ set $v_j = 3$. For all other y_j set $v_j = 1$. For each vertex $y_i \in Y$, map y_i to $w = (w_0, w_1, \ldots, w_n)$ as follows. Set $w_0 = 2$. Set $v_i = 4$ and for all other v_j set $v_j = 2$.

Consider any x_i, y_j. Note that $v_0 = 1, w_0 = 2$, so these coordinates are adjacent. Now observe that for all $k \geq 1, w_k \in \{2, 4\}$ and $v_k \in \{1, 3\}$, so v_k, w_k are not adjacent only if $v_k = 1, w_k = 4$. In that case, $j = k$, and since $v_k = 1$ it must be that $y_k = y_j \notin N(x_i)$, so x_i, y_j are not adjacent. Finally, note that any distinct x_i, x_j are not mapped to adjacent vertices since both have $v_0 = 1$, and for the same reason any distinct y_i, y_j are also not mapped to adjacent vertices. □

Proof of statement 3. Let $X_1, \ldots, X_t, Y_1, \ldots, Y_t$ partition the vertices of G so that $G[X_i, Y_i]$ is a biclique for each $i \in [t]$ and $G[X_i, Y_j]$ is an independent set for each $i \neq j$. (Note that we may have $Y_t = \emptyset$, in which case X_t is simply an independent set).

Let $H \in \mathsf{cl}(G^{\times})$ so $H \sqsubset G^{\times d}$ for some d. For each sequence $s = (s_1, \ldots, s_d)$ with $s_i \in [t]$, define the pair (A_s, B_s) where

$$A_s = \{x \in V(G^{\times d}) : \forall i \in [t], x_i \in X_{s_i}\}$$
$$B_s = \{x \in V(G^{\times d}) : \forall i \in [t], x_i \in Y_{s_i}\}\,.$$

Since the sets X_i, Y_i partition $V(G)$, the sets A_s, B_s partition $V(G^{\times d})$. We claim that each A_s and B_s is an independent set, that $G^{\times d}[A_s, B_s]$ is a complete bipartite graph for each $s \in [t]^d$, and that for every $s \neq s'$, the graphs $G^{\times d}[A_s, B_{s'}]$, $G^{\times d}[A_s, A_{s'}]$, and $G^{\times d}[B_s, B_{s'}]$ are independent sets. This suffices to show that $G^{\times d}$ is a bipartite equivalence graph.

Suppose $x, y \in A_s$. Then x_i is not adjacent to y_i for any $i \in [d]$ since each X_{s_i} is an independent set in G. The same argument holds for $x, y \in B_s$.

For any $x \in A_s, y \in B_s$, it holds that for each $i \in [d]$, $x_i \in X_{s_i}$ and $y_i \in Y_{s_i}$, so x_i, y_i are adjacent in G. Therefore x, y are adjacent in $G^{\times d}$. On the other hand, if $s \neq s'$, then for $x \in A_s$ and $y \in B_{s'}$, there is some coordinate $i \in [d]$ such that $s_i \neq s'_i$. But then $x_i \in X_{s_i}$ and $y_i \in Y_{s'_i}$ and x_i, y_i are not adjacent in G, so x, y are not adjacent. A similar argument holds for $x \in A_s, y \in A_{s'}$ and $x \in B_s, y \in B_{s'}$. □

B.3 Lexicographic Product

Any graph G is either an independent set or clique, or it contains one of P_3 or $K_2 + K_1$ as an induced subgraph. Therefore the following lemma suffices to verify Conjecture 1.2 for the families $\mathsf{cl}(G^\diamond)$.

Lemma B.5.

(1) *If G is an independent set or a clique then $\mathsf{cl}(G^\diamond)$ is the family of independent sets or the family of cliques.*

(2) $\mathcal{C}^{\bullet\circ} \subset \mathsf{cl}(P_3^\diamond)$.

(3) $\mathcal{C}^{\bullet\circ} \subset \mathsf{cl}((K_2 + K_1)^\diamond)$.

Proof of statement 1. Trivial. □

Proof of statement 2. Identify the vertices of P_3 with the numbers $[3]$ so that $(1,2),(2,3)$ are the edges of P_3. Let $\{x_1,\dots,x_n,y_1,\dots,y_n\}$ be the vertices of $H_n^{\bullet\circ}$ so that every distinct pair (x_i,x_j) is adjacent, every distinct pair (y_i,y_j) is non-adjacent, and (x_i,y_j) are adjacent if and only if $i \le j$. Map each x_i to the vertex $v = (v_1,\dots,v_{n+1}) \in P_3^{\diamond(n+1)}$ defined by setting $v_k = 1$ when $k \le i$ and $v_k = 2$ when $k > i$. Map each y_i to the vertex $w = (w_1,\dots,w_{n+1}) \in P_3^{\diamond(n+1)}$ defined by setting $w_k = 1$ when $k \le i$ and $w_k = 3$ when $k > i$.

Consider two vertices x_i, y_j. If $i \le j$ then $v_k = w_k = 1$ for all $k \le i \le j$ while $v_k = 2$ for all $k > i$. Assuming $i < j$, for $i < k \le j$ it holds that $v_k = 2, w_k = 1$, so v,w are adjacent since v_{i+1}, w_{i+1} are adjacent. Assuming $i = j$, since $i,j < n+1$ then it holds that $v_{i+1} = 2, w_{i+1} = 3$, so v,w are adjacent.

Now suppose that $i > j$. Then $v_k = w_k = 1$ for all $k \le j$. But $v_{j+1} = 1$ while $w_{j+1} = 3$ so v_{j+1}, w_{j+1} are not adjacent in P_3. So v,w are not adjacent.

Finally, it is easy to verify that each distinct pair x_i,x_j is mapped to adjacent vertices and each distinct pair y_i,y_j is mapped to non-adjacent vertices in $P_3^{\diamond(n+1)}$. □

Proof of statement 3. Using the same mapping as above with vertex 3 being the isolated vertex of $K_2 + K_1$ one can check that $H_{n-1}^{\bullet\circ}$ is an induced subgraph of $(K_2+K_1)^{\diamond(n+1)}$. □

C Bibliographic Remark on Greater-Than

Recall the lower bound for Greater-Than:

Theorem C.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.

However, as noted in a CSTheory StackExchange question of Sasho 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 Amit 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 C.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 C.3 ([MNSW98])**.** *Any public-coin randomized one-way communication protocol for Augmented-Index requires $\Omega(k)$ bits of communication.*

Proof of Theorem C.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 C.3, the communication cost of Greater-Than for $n = 2^k$ is at least $\Omega(k) = \Omega(\log n)$. ☐

Acknowledgments

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

References

[ABR05] Stephen Alstrup, Philip Bille, and Theis Rauhe. Labeling schemes for small distances in trees. *SIAM Journal on Discrete Mathematics*, 19(2):448–462, 2005.

[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.

[ACLZ15] Aistis Atminas, Andrew Collins, Vadim Lozin, and Victor Zamaraev. Implicit representations and factorial properties of graphs. *Discrete Mathematics*, 338(2):164–179, 2015.

[AHKO21] Jungho Ahn, Kevin Hendrey, Donggyu Kim, and Sang-il Oum. Bounds for the twin-width of graphs. *arXiv preprint arXiv:2110.03957*, 2021.

[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.

[BBG14] Eric Blais, Joshua Brody, and Badih Ghazi. The information complexity of hamming distance. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[BBM+20] Alexander R. Block, Simina Branzei, Hemanta K. Maji, Himanshi Mehta, Tamalika Mukherjee, and Hai H. Nguyen. P_4-free partition and cover numbers and application. Cryptology ePrint Archive, Report 2020/1605, 2020. https://ia.cr/2020/1605.

[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.

[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 Hliněnỳ. Twin-width is linear in the poset width. *arXiv preprint arXiv:2106.15337*, 2021.

[BKTW20] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width i: tractable fo 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.

[Cha18] Maurice Chandoo. A complexity theory for labeling schemes. *arXiv preprint arXiv:1802.02819*, 2018.

[Cha20] Amit Chakrabarti. One-way randomized communication complexity of greater-than. Theoretical Computer Science Stack Exchange, 2020. URL: https://cstheory.stackexchange.com/q/48110 (version: 2020-12-30).

[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 Erdös-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.

[ES35] Paul Erdös and George Szekeres. A combinatorial problem in geometry. *Compositio mathematica*, 2:463–470, 1935.

[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.

[GPT21] Jakub Gajarskỳ, Michał Pilipczuk, and Szymon Toruńczyk. Stable graphs of bounded twin-width. *arXiv preprint arXiv:2107.03711*, 2021.

[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.

[HHH21] Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. Dimension-free bounds and structural results in communication complexity. *Electron. Colloquium Comput. Complex.*, page 66, 2021.

[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.

[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 Nešetšil, 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.

[LZ15] Vadim V Lozin and Victor Zamaraev. Boundary properties of factorial classes of graphs. *Journal of Graph Theory*, 78(3):207–218, 2015.

[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.

[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).

[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.

[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.

[SS21] André Schidler and Stefan Szeider. A SAT approach to twin-width. *arXiv preprint arXiv:2110.06146*, 2021.

[ST21] Pierre Simon and Szymon Toruńczyk. Ordered graphs of bounded twin-width. *arXiv preprint arXiv:2102.06881*, 2021.

[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.