

# Universal Graph Compression: Stochastic Block Models

Alankrita Bhatt\*

University of California San Diego  
La Jolla, CA 92093, USA  
a2bhatt@eng.ucsd.edu

Ziao Wang\*

University of British Columbia  
Vancouver, BC V6T1Z4, Canada  
ziaow@ece.ubc.ca

Chi Wang

Microsoft Research, Redmond  
Redmond, WA 98052, USA  
wang.chi@microsoft.com

Lele Wang

University of British Columbia  
Vancouver, BC V6T1Z4, Canada  
lelewang@ece.ubc.ca

## Abstract

Motivated by the prevalent data science applications of processing and mining large-scale graph data such as social networks, web graphs, and biological networks, as well as the high I/O and communication costs of storing and transmitting such data, this paper investigates lossless compression of data appearing in the form of a labeled graph. In particular, we consider a widely used random graph model, stochastic block model (SBM), which captures the clustering effects in social networks. An information-theoretic *universal compression* framework is applied, in which one aims to design a *single* compressor that achieves the asymptotically optimal compression rate, for every SBM distribution, without knowing the parameters of the SBM that generates the data. Such a graph compressor is proposed in this paper, which universally achieves the optimal compression rate for a wide class of SBMs with edge probabilities ranging from  $O(1)$  to  $\Omega(1/n^{2-\epsilon})$  for any  $0 < \epsilon < 1$ .

Existing universal compression techniques are developed mostly for *stationary ergodic* one-dimensional sequences with *fixed* alphabet size and entropy *linear* in the number of variables. However, the adjacency matrix of SBM has complex two-dimensional correlations and sublinear entropy in the sparse regime. These challenges are alleviated through a carefully designed transform that converts two-dimensional correlated data into *almost* i.i.d. blocks. The blocks are then compressed by a standard Krichevsky–Trofimov compressor, whose length analysis is generalized to identically distributed but arbitrarily correlated sequences with slowly growing alphabet size and sublinear entropy. In four benchmark graph datasets (protein-to-protein interaction, LiveJournal friendship, Flickr, and YouTube), the compressed files from competing algorithms (including CSR, Ligr+, PNG image compressor, and Lempel–Ziv compressor for two-dimensional data) take 2.4 to 27 times the space needed by the proposed scheme.

---

\*Alankrita Bhatt and Ziao Wang contributed equally to this work.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Setup . . . . .	3
1.2	Main Results . . . . .	4
<b>2</b>	<b>Algorithm: Universal Graph Compressor</b>	<b>4</b>
<b>3</b>	<b>Main Ideas in Establishing Universality</b>	<b>7</b>
<b>4</b>	<b>Proof of Universality</b>	<b>9</b>
4.1	Graph Entropy . . . . .	9
4.2	Asymptotic i.i.d. via Block Decomposition . . . . .	10
4.3	Length of the Laplace Probability Assignment . . . . .	13
4.4	Length of the KT probability assignment . . . . .	14
4.5	Proof of Theorem 2 . . . . .	16
<b>5</b>	<b>Second order analysis in the sparse regime</b>	<b>17</b>
5.1	Basic definitions on rooted graphs . . . . .	17
5.2	Local weak convergence . . . . .	18
5.3	BC entropy . . . . .	20
5.4	Achieving BC entropy in the sparse regime . . . . .	20
<b>6</b>	<b>Stationarity in the stochastic block model</b>	<b>21</b>
<b>7</b>	<b>Experiments</b>	<b>23</b>

# 1 Introduction

In many data science applications, data appears in the form of large-scale graphs. For example, in social networks, vertices represent users and an edge between vertices represents friendship; in the World Wide Web, vertices are websites and edges indicate the hyperlinks from one site to the other; in biological systems, vertices can be proteins and edges illustrate protein-to-protein interaction. Such graphs may contain billions of vertices. In addition, edges tend to be correlated with each other since, for example, two people sharing many common friends are likely to be friends as well. How to efficiently compress such large-scale structural information to reduce the I/O and communication costs in storing and transmitting such data is a persisting challenge in the era of big data.

The literature on graph compression is vast. Existing compression schemes follow various different methodologies. Several methods exploited combinatorial properties such as cliques and cuts in the graph [1, 2]. Many works targeted at domain-specific graphs such as web graphs [3], biology networks [4, 5], and social network graphs [6]. Various representations of graphs were proposed, such as the text-based method, where the neighbor list of each vertex is treated as a “word” [7, 8], and the  $k^2$ -tree method, where the adjacency matrix is recursively partitioned into  $k^2$  equal-size submatrices [9]. *Succinct* graph representations that enable certain types of fast computation, such as adjacency query or vertex degree query, were also widely studied [10]. While most compression schemes are for labeled graphs, there are also works considering lossless compression of unlabeled graphs [11–13], graphs with marks on its edges and vertices [14–16], or (correlated) data on the graph [17, 18]. We refer the readers to [19] for an exhaustive survey on lossless graph compression and space-efficient graph representations.

In this paper, we take an information theoretic approach to study lossless compression of a graph. We assume the graph is generated by some random graph model and investigate lossless compression schemes that achieve the theoretical limit, i.e., the entropy of the graph, asymptotically as the number of vertices goes to infinity. When the underlying distribution/statistics of the random graph model is known, optimal lossless compression can be achieved by methods like Huffman coding. However, in most real-world applications, the exact distribution is usually hard to obtain and the data we are given is a single realization of this distribution. This motivates us to consider the framework of *universal compression*, in which we assume the underlying distribution belongs to a known family of distributions and require that the encoder and the decoder should not be a function of the underlying distribution. The goal of universal compression is to design a single compression scheme that universally achieves the optimal theoretical limit, for every distribution in the family, without knowing which distribution generates the data. For this paper, we focus on the family of *stochastic block models*, which are widely used random graph models that capture the clustering effect in social networks. Our goal is to develop a universal graph compression scheme for a family of stochastic block models with as wide range of parameters as possible.

How to design computationally efficient universal compression scheme is a fundamental question in information theory. In the past several decades, a large number of universal compressors were proposed for one-dimensional sequences with fixed alphabet size, whose entropy is linear in the number of variables. Prominent results include the Laplace and Krichevsky–Trofimov (KT) compressors for i.i.d. processes [20, 21], Lempel–Ziv compressor [22, 23] and Burrows–Wheeler transform [24] for stationary ergodic processes, and context tree weighting [25] for finite memory processes. Many of these have been adopted in standard data compression applications such as `compress`, `gzip`, GIF, TIFF, and `bzip2`. Despite these exciting developments, existing universal compression techniques fall short of establishing optimality results for graph data due to the following challenges. Firstly, graph data generated from a stochastic block model has non-stationary two-dimensional correlation,

so existing techniques do not immediately apply here. Secondly, in many practical applications, where the graph is sparse, the entropy of the graph may be sublinear in the number of entries in the adjacency matrix.

For the first challenge, a natural question arising is: can we convert the two-dimensional adjacency matrix of the graph into a one-dimensional sequence in some order and apply a universal compressor for the sequence? For some simple graph model such as Erdős–Rényi graph, where each edge is generated i.i.d. with probability  $p$ , this would indeed work. For more complex graph models including stochastic block models, it is unclear whether there is an ordering of the entries that results in a stationary process. We will show in Section 6 several orders including row-by-row, column-by-column, and diagonal-by-diagonal fail to produce a stationary process. We alleviate this challenge by designing a decomposition of the adjacency matrix into blocks. We then show in Theorem 3 that with a carefully chosen parameter, the block decomposition converts two-dimensional correlated entries into a sequence of *almost* i.i.d. blocks with slowly growing alphabet size. To address the second challenge, we adjust the standard definition of universality, which normalizes the compression length by the number of variables. The new definition of universality accommodates data with unknown leading order in its entropy expression.

Lossless compression for stochastic block models was first studied by Abbe [17] (albeit not under the universal compression framework). The focus there is two-fold: 1) compute the entropy of the stochastic block model; 2) explore the relation between community detection and compression. Several interesting questions were presented: Knowing the community assignments will help compression since edges can be grouped into i.i.d. subsets. But is community detection necessary for compression? In the regime when community detection is not possible, how do we compress the graph? We answer these questions in this paper by presenting a universal compressor that does not require knowledge of the edge probabilities, the community assignments, or the number of communities. Our compressor remains universal even in the regime when community detection is information theoretically impossible. As a consequence, universal compression is a fundamentally easier task than community detection for stochastic block models.

Recently, universal compression of graphs with marked edges and vertices is studied by Delgosha and Anantharam [16, 26]. They focus on the *sparse* graph regime, where the number of edges is in the same order as the number of vertices  $n$ . They employ the framework of local weak convergence, which provides a technique to view a sequence of graphs as a sequence of distributions on neighbourhood structures. Built on this framework, they propose an algorithm that compresses graphs by describing the local neighbourhood structures. Moreover, they introduce a universality/optimalty criterion through a notion of entropy for graph sequences under the local weak convergence framework, known as the *BC entropy* [27]. This universality criterion is stronger than the one used in this paper. It requires the asymptotic length of the compressor to match the constants in both first and second order terms in Shannon entropy, whereas the universality criterion we use only requires to match the first order term. As a consequence of the stronger criterion, the compressor in [26] is universal over a smaller random graph family. In comparison, we expand the range of edge numbers from  $\Theta(n)$  in the sparse regime to  $\Theta(n^\alpha)$  for every  $0 < \alpha \leq 2$  and propose a single universal compressor for the whole family under the weaker universality criterion. In Section 5, we evaluate the proposed compressor under the criterion in [26] for the family of *symmetric* SBMs. The proposed compressor achieves a similar performance in terms of BC entropy in the sparse regime.

The rest of the paper is organized as follows. In Section 1.1, we define universality over a family of graph distributions and the stochastic block models. We present our main result in Section 1.2, which is a graph compressor that is universal for a family containing most non-trivial stochastic block models. We describe the proposed graph compressor in Section 2. We illustrate key steps in

establishing universality in Section 3 and elaborate the proof of each step in Section 4. In Section 5, we provide the second order analysis of the expected length of our compressor and compare it to the one in [26]. In Section 6, we explain why existing universal compressors developed for stationary processes may not be immediately applicable for some one-dimensional ordering of entries in the adjacency matrix. In Section 7, we implement our compressor in four benchmark graph datasets and compare its empirical performance to four competing algorithms.

**Notation.** For an integer  $n$ , let  $[n] = \{1, 2, \dots, n\}$ . Let  $\log(\cdot) = \log_2(\cdot)$ . We follow the standard order notation:  $f(n) = O(g(n))$  if  $\lim_{n \rightarrow \infty} \frac{|f(n)|}{g(n)} < \infty$ ;  $f(n) = \Omega(g(n))$  if  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} > 0$ ;  $f(n) = \Theta(g(n))$  if  $f(n) = O(g(n))$  and  $f(n) = \Omega(g(n))$ ;  $f(n) = o(g(n))$  if  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$ ;  $f(n) = \omega(g(n))$  if  $\lim_{n \rightarrow \infty} \frac{|f(n)|}{g(n)} = \infty$ ; and  $f(n) \sim g(n)$  if  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1$ .

## 1.1 Problem Setup

For simplicity, we focus on simple (undirected, unweighted, no self-loop) graphs with labeled vertices in this paper. But our compression scheme and the corresponding analysis can be extended to more general graphs. Let  $\mathcal{A}_n$  be the set of all labeled simple graphs on  $n$  vertices. Let  $\{0, 1\}^i$  be the set of binary sequences of length  $i$ , and set  $\{0, 1\}^* = \cup_{i=0}^{\infty} \{0, 1\}^i$ . A lossless graph compressor  $C: \mathcal{A}_n \rightarrow \{0, 1\}^*$  is a one-to-one function that maps a graph to a binary sequence. Let  $\ell(C(A_n))$  denote the length of the output sequence. When  $A_n$  is generated from a distribution, it is known that the entropy  $H(A_n)$  is a fundamental lower bound on the expected length of any lossless compressor [29, Theorem 8.3]

$$H(A_n) - \log(e(H(A_n) + 1)) \leq \mathbb{E}[\ell(C(A_n))], \quad (1)$$

and therefore

$$\liminf_{n \rightarrow \infty} \frac{\mathbb{E}[\ell(C(A_n))]}{H(A_n)} \geq 1.$$

Thus, a graph compressor is said to be *universal* for the family of distributions  $\mathcal{P}$  if for all distribution  $P \in \mathcal{P}$  and  $A_n \sim P$ , we have

$$\limsup_{n \rightarrow \infty} \frac{\mathbb{E}[\ell(C(A_n))]}{H(A_n)} = 1. \quad (2)$$

A stochastic block model  $\text{SBM}(n, L, \mathbf{p}, \mathbf{W})$  defines a probability distribution over  $\mathcal{A}_n$ . Here  $n$  is the number of vertices,  $L$  is the number of communities. Each vertex  $i \in [n]$  is associated with a community assignment  $X_i \in [L]$ . The length- $L$  column vector  $\mathbf{p} = (p_1, p_2, \dots, p_L)^T$  is a probability distribution over  $[L]$ , where  $p_i$  indicates the probability that any vertex is assigned community  $i$ .  $\mathbf{W}$  is an  $L \times L$  symmetric matrix, where  $W_{ij}$  represents the probability of having an edge between a vertex with community assignment  $i$  and a vertex with community assignment  $j$ . We say  $A_n \sim \text{SBM}(n, L, \mathbf{p}, \mathbf{W})$  if the community assignments  $X_1, X_2, \dots, X_n$  are generated i.i.d. according to  $\mathbf{p}$  and for every pair  $1 \leq i < j \leq n$ , an edge is generated between vertex  $i$  and vertex  $j$  with probability  $W_{X_i, X_j}$ . In other words, in the adjacency matrix  $A_n$  of the graph,  $A_{ij} \sim \text{Bern}(W_{X_i, X_j})$  for  $i < j$ ; the diagonal entries  $A_{ii} = 0$  for all  $i \in [n]$ ; and  $A_{ij} = A_{ji}$  for  $i > j$ . We assume all the entries in  $\mathbf{W}$  are in the same regime  $f(n)$  and write  $\mathbf{W} = f(n)\mathbf{Q}$ , where  $\mathbf{Q}$  is an  $L \times L$  symmetric matrix with constant entries  $Q_{ij} = \Theta(1)$  for all  $i, j \in [L]$ . We assume all entries

in  $\mathbf{p}$  are  $\Theta(1)$ . We will consider two families of stochastic block models: For  $0 < \epsilon < 1$ ,

$$\mathcal{P}_1(\epsilon): \text{SBM with } L = \Theta(1), f(n) = O(1), f(n) = \Omega\left(\frac{1}{n^{2-\epsilon}}\right), \quad (3)$$

$$\mathcal{P}_2(\epsilon): \text{SBM with } L = \Theta(1), f(n) = o(1), f(n) = \Omega\left(\frac{1}{n^{2-\epsilon}}\right). \quad (4)$$

Note that the edge probability  $\frac{1}{n^2}$  is the threshold for a random graph to contain an edge with high probability [30]. Thus, the family  $\mathcal{P}_1(\epsilon)$  covers most non-trivial SBM graphs. Clearly,  $\mathcal{P}_2(\epsilon)$  is a strict subset of  $\mathcal{P}_1(\epsilon)$ , as it does not contain the constant regime  $f(n) = 1$ .

## 1.2 Main Results

The main contribution of this paper is providing two compressors universal over the classes  $\mathcal{P}_1(\epsilon)$  and  $\mathcal{P}_2(\epsilon)$  respectively for  $0 < \epsilon < 1$ . Note that a compressor universal over the class  $\mathcal{P}_1(\epsilon)$  is also universal over the class  $\mathcal{P}_2(\epsilon)$ , but our compressor designed specifically for the class  $\mathcal{P}_2(\epsilon)$  has a lower computational complexity. We will formally state the results in the next two theorems.

**Theorem 1** (Universality over  $\mathcal{P}_1$ ). *For every  $0 < \epsilon < 1$ , the graph compressor  $C_k$  defined in Section 2 is universal over the family  $\mathcal{P}_1(\epsilon)$  provided that*

$$0 < \delta < \epsilon, \quad k \leq \sqrt{\delta \log n}, \quad \text{and} \quad k = \omega(1).$$

**Theorem 2** (Universality over  $\mathcal{P}_2$ ). *For every  $0 < \epsilon < 1$ , the graph compressor  $C_1$  defined in Section 2 is universal over the family  $\mathcal{P}_2(\epsilon)$ .*

For now, one can think of  $k$  as a parameter that defines a compression scheme  $C_k$ —the exact definition will become clear in the next section when we precisely define the compressors.

## 2 Algorithm: Universal Graph Compressor

In this section, we describe our universal graph compression scheme. For each  $k$  that divides  $n$ , the graph compressor  $C_k: \mathcal{A}_n \rightarrow \{0, 1\}^*$  is defined as follows.

- **Block decomposition.** Let  $n' = \frac{n}{k}$ . For  $1 \leq i, j \leq n'$ , let  $\mathbf{B}_{ij}$  be the submatrix of  $A_n$  formed by the rows  $(i-1)k+1, (i-1)k+2, \dots, ik$  and the columns  $(j-1)k+1, (j-1)k+2, \dots, jk$ . For example, we have

$$\mathbf{B}_{12} = \begin{bmatrix} A_{1,k+1} & A_{1,k+2} & \cdots & A_{1,2k} \\ A_{2,k+1} & A_{2,k+2} & \cdots & A_{2,2k} \\ \vdots & \vdots & \ddots & \vdots \\ A_{k,k+1} & A_{k,k+2} & \cdots & A_{k,2k} \end{bmatrix}. \quad (5)$$

We then write  $A_n$  in the block-matrix form as

$$A_n = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \cdots & \mathbf{B}_{1,n'} \\ \mathbf{B}_{21} & \mathbf{B}_{22} & \cdots & \mathbf{B}_{2,n'} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{n',1} & \mathbf{B}_{n',2} & \cdots & \mathbf{B}_{n',n'} \end{bmatrix}. \quad (6)$$

Denote

$$\mathbf{B}_{\text{ut}} := \mathbf{B}_{12}, \mathbf{B}_{13}, \mathbf{B}_{23}, \mathbf{B}_{14}, \mathbf{B}_{24}, \mathbf{B}_{34}, \dots, \mathbf{B}_{1,n'}, \dots, \mathbf{B}_{n'-1,n'} \quad (7)$$

as the sequence of off-diagonal blocks in the upper triangle and

$$\mathbf{B}_d := \mathbf{B}_{11}, \mathbf{B}_{22}, \dots, \mathbf{B}_{n',n'} \quad (8)$$

as the sequence of diagonal blocks.

- **Binary to  $m$ -ary conversion.** Let  $m := 2^{k^2}$ . Each  $k \times k$  block with binary entries in the two block sequences  $\mathbf{B}_{\text{ut}}$  and  $\mathbf{B}_d$  is converted into a symbol in  $[m]$ .
- **KT probability assignment.** Apply KT sequential probability assignment for the two  $m$ -ary sequences  $\mathbf{B}_{\text{ut}}$  and  $\mathbf{B}_d$  respectively. Given an  $m$ -ary sequence  $x_1, x_2, \dots, x_N$ , *KT sequential probability assignment* defines  $N$  conditional probability distributions over  $[m]$  as follows. For  $j = 0, 1, 2, \dots, N - 1$ , assign conditional probability

$$q_{\text{KT}}(i|x^j) := q_{\text{KT}}(X_{j+1} = i | X^j = x^j) = \frac{N_i(x^j) + 1/2}{j + m/2} \quad \text{for each } i \in [m], \quad (9)$$

where  $X^j := (X_1, \dots, X_j)$ ,  $x^j := (x_1, x_2, \dots, x_j)$ , and  $N_i(x^j) := \sum_{k=1}^j \mathbb{1}\{x_k = i\}$  counts the number of symbol  $i$  in  $x^j$ .

- **Adaptive arithmetic coding.** With the KT sequential probability assignments, compress the two sequences  $\mathbf{B}_{\text{ut}}$  and  $\mathbf{B}_d$  separately using adaptive arithmetic coding [31] (see description in Algorithm 1). In case  $k = 1$ , the diagonal sequence  $\mathbf{B}_d$  becomes an all-zero sequence since we assume the graph is simple. So we will only compress the off-diagonal sequence  $\mathbf{B}_{\text{ut}}$ .

---

**Algorithm 1:**  $m$ -ary adaptive arithmetic encoding with KT probability assignment

---

**Input** : Data sequence  $x^N$ , alphabet size  $m$

Initialize **lower** = 0, **upper** = 1, **logprob** = 0,  $N_1 = N_2 = \dots = N_m = 0$ ;

**for**  $j = 0, 1, \dots, N - 1$  **do**

	<b>range</b> $\leftarrow$ <b>upper</b> - <b>lower</b> ;
	<b>for</b> $i = 1, 2, \dots, x_{j+1}$ <b>do</b>
	Compute $q_{\text{KT}}(i x^j) = \frac{N_i+1/2}{j+m/2}$ ;
	<b>upper</b> $\leftarrow$ <b>lower</b> + <b>range</b> $\cdot \sum_{i=1}^{x_{j+1}} q_{\text{KT}}(i x^j)$ ;
	<b>lower</b> $\leftarrow$ <b>upper</b> - <b>range</b> $\cdot q_{\text{KT}}(x_{j+1} x^j)$ ;
	$N_{x_{j+1}} \leftarrow N_{x_{j+1}} + 1$ ;
	<b>logprob</b> $\leftarrow$ <b>logprob</b> + $\log(q_{\text{KT}}(x_{j+1} x^j))$ ;

**Output:** the binary representation of  $\frac{1}{2}(\text{lower} + \text{upper})$  with  $\lceil -\text{logprob} \rceil + 1$  bits

---

Given the compressed graph sequence  $y^L$ , the number of vertices  $n$  and the block size  $k$ , the graph decompressor  $D_k: \{0, 1\}^* \rightarrow \mathcal{A}_n$  is defined as follows.

- **Adaptive arithmetic decoding.** With the KT sequential probability assignments defined in (9), decompress the two code sequences for  $\mathbf{B}_{\text{ut}}$  and  $\mathbf{B}_d$  separately using adaptive arithmetic decoding (see Algorithm 2). The length of data sequence  $\mathbf{B}_{\text{ut}}$  and  $\mathbf{B}_d$  are  $\frac{n}{k}(\frac{n}{k} - 1)/2$  and  $\frac{n}{k}$  respectively.
- **$m$ -ary to binary conversion.** Each  $m$ -ary symbol in the sequence is converted to a  $k^2$ -bit binary number and further converted into a  $k \times k$  block with binary entries.

---

**Algorithm 2:**  $m$ -ary adaptive arithmetic decoding with KT probability assignment

---

**Input** : Binary sequence  $y^L$ , alphabet size  $m = 2^{k^2}$ , length of data sequence  $N$

Add ‘0.’ before sequence  $y^L$  and convert it into a decimal real number  $Y$ . Initialize

$\text{lower} = 0, \text{upper} = 1, N_1 = N_2 = \dots = N_m = 0;$

**for**  $j = 0, 1, \dots, N - 1$  **do**

$\text{range} \leftarrow \text{upper} - \text{lower};$

**for**  $i = 1, 2, \dots, m$  **do**

        Compute  $q_{\text{KT}}(i|x^j) = \frac{N_i+1/2}{j+m/2};$

    Find minimum  $z \in [m]$  such that  $\text{lower} + \text{range} \cdot \sum_{i=1}^z q_{\text{KT}}(i|x^j) > Y;$

$\text{upper} \leftarrow \text{lower} + \text{range} \cdot \sum_{i=1}^z q_{\text{KT}}(i|x^j);$

$\text{lower} \leftarrow \text{upper} - \text{range} \cdot q_{\text{KT}}(z|x^j);$

$N_z \leftarrow N_z + 1;$

$x_{j+1} \leftarrow z;$

**Output:** the  $m$ -ary data sequence  $x_1, x_2, \dots, x_N$

---

- **Adjacency matrix recovery.** With the blocks in  $\mathbf{B}_{\text{ut}}$  and  $\mathbf{B}_{\text{d}}$ , recover the adjacency matrix of  $A_n$  in the order described in (6), (7), and (8).

One can check that  $C_k$  is well-defined. The block decomposition and the binary to  $m$ -ary conversion are clearly one-to-one. It is also known that for any valid probability assignment, arithmetic coding produces a prefix code, which as also one-to-one.

The computational complexity of the proposed algorithm is  $O(2^{k^2} n^2)$ . For the choice of  $k$  that achieves universality over  $\mathcal{P}_1(\epsilon)$  family in Theorem 1,  $O(2^{k^2} n^2) = O(n^{2+\delta})$  for  $\delta < \epsilon$ . For the choice of  $k$  that achieves universality over  $\mathcal{P}_2(\epsilon)$  family in Theorem 2,  $O(2^{k^2} n^2) = O(n^2)$ .

The orders in  $\mathbf{B}_{\text{ut}}$  and  $\mathbf{B}_{\text{d}}$  do not matter in terms of establishing universality. The current orders in (7) and (8) together with arithmetic coding enable a *horizon free* implementation. That is, the encoder does not need to know the *horizon*  $n$  to start processing the data and can output partial coded bits *on the fly* before receiving all the data. This leads to short encoding and decoding delay. For some real-world applications, for example, when the number of users increases in a large social network, this compressor has the advantage of not requiring to re-process existing data and re-compress the whole graph from scratch.

**Remark 1 (Laplace probability assignment).** As an alternative to the KT sequential probability assignment, one can also use the Laplace sequential probability assignment. Given an  $m$ -ary sequence  $x_1, x_2, \dots, x_N$ , *Laplace sequential probability assignment* defines  $N$  conditional probability distributions over  $[m]$  as follows. For  $j = 0, 1, 2, \dots, N - 1$ , we assign conditional probability

$$q_{\text{L}}(X_{j+1} = i | X^j = x^j) = \frac{N_i(x^j) + 1}{j + m} \quad \text{for each } i \in [m]. \quad (10)$$

Both methods can be shown to be universal, while Laplace probability assignment has a much cleaner derivation. However, KT probability assignment produces a better empirical performance. For this reason, we keep both in the paper.

### 3 Main Ideas in Establishing Universality

In this section, we establish the universality of the graph compressor in Section 2.

**Graph Entropy** We first calculate the entropy of the (random) graph  $A_n$ , which, recall, is the fundamental lower bound on the expected compression length for any compression scheme. Since to establish optimality we need to show that  $\limsup_{n \rightarrow \infty} \frac{\mathbb{E}[\ell(C(A_n))]}{H(A_n)} \leq 1$ , we will only be concerned with the first order term in  $H(A_n)$ .

**Lemma 1** (Graph entropy). *Let  $A_n \sim SBM(n, L, \mathbf{p}, f(n)\mathbf{Q})$  with  $f(n) = O(1)$ ,  $f(n) = \Omega(\frac{1}{n^2})$ , and  $L = \Theta(1)$ . For  $0 \leq p \leq 1$ , let  $h(p) \triangleq -p \log(p) - (1-p) \log(1-p)$  denote the binary entropy function. For a matrix  $W$  with entries in  $[0, 1]$ , let  $h(W)$  be a matrix of the same dimension whose  $(i, j)$  entry is  $h(W_{ij})$ . Then*

$$H(A_n) = \binom{n}{2} H(A_{12} | X_1, X_2) (1 + o(1)) \quad (11)$$

$$= \binom{n}{2} \mathbf{p}^T h(f(n)\mathbf{Q}) \mathbf{p} + o(n^2 h(f(n))). \quad (12)$$

In particular, when  $f(n) = \Omega(\frac{1}{n^2})$  and  $f(n) = o(1)$ , expression (12) can be further simplified as

$$H(A_n) = \binom{n}{2} f(n) \log\left(\frac{1}{f(n)}\right) (\mathbf{p}^T \mathbf{Q} \mathbf{p} + o(1)). \quad (13)$$

**Remark 2.** In the regime  $f(n) = \Omega(\frac{1}{n})$  and  $f(n) = O(1)$ , the above result has been established in [17]. We extend the analysis to the regime  $f(n) = o(\frac{1}{n})$  and  $f(n) = \Omega(\frac{1}{n^2})$ .

**Remark 3.** Lemma 1 can be used to calculate the entropy of the graph for certain important regimes of  $f(n)$ , in which the SBM displays characteristic behavior. For  $f(n) = 1$ , we have  $H(A_n) = \binom{n}{2} h(\mathbf{p}^T \mathbf{Q} \mathbf{p}) (1 + o(1))$ ; for  $f(n) = \frac{\log n}{n}$  (the regime where the phase transition for exact recovery of the community assignments occurs [32, 33]), we have  $H(A_n) = \frac{n \log^2 n}{2} (\mathbf{p}^T \mathbf{Q} \mathbf{p} + o(1))$ ; when  $f(n) = \frac{1}{n}$  (the regime where the phase transition for detection between SBM and the Erdős–Rényi model occurs [34]), we have  $H(A_n) = \frac{n \log n}{2} (\mathbf{p}^T \mathbf{Q} \mathbf{p} + o(1))$ ; when  $f(n) = \frac{1}{n^2}$  (the regime where the phase transition for the existence of an edge occurs), we have  $H(A_n) = \log n (\mathbf{p}^T \mathbf{Q} \mathbf{p} + o(1))$ .

**Asymptotic i.i.d. via Block Decomposition** To compress the matrix  $A_n$ , we wish to decompose it into a large number of components that have little correlation between them. This leads to the idea of block decomposition described previously. Since the sequence of blocks are used to compress  $A_n$ , the next theorem claims these blocks are identically distributed and asymptotically independent in a precise sense described as follows.

**Theorem 3** (Block decomposition). *Let  $A_n \sim SBM(n, L, \mathbf{p}, f(n)\mathbf{Q})$  with  $f(n) = \Omega(\frac{1}{n^{2-\epsilon}})$  for some  $0 < \epsilon < 1$ ,  $f(n) = O(1)$ , and  $L = \Theta(1)$ . Let  $k$  be an integer that divides  $n$  and  $n' = n/k$ . Consider the  $k \times k$  block decomposition in (6). We have all the off-diagonal blocks share the same joint distribution; all the diagonal blocks share the same joint distribution. In other words, for any  $1 \leq i_1, i_2, j_1, j_2 \leq n'$  with  $i_1 \neq j_1, i_2 \neq j_2$  and  $1 \leq l_1, l_2 \leq n'$ , we have*

$$\mathbf{B}_{i_1, j_1} \stackrel{d}{=} \mathbf{B}_{i_2, j_2},$$

$$\mathbf{B}_{l_1, l_1} \stackrel{d}{=} \mathbf{B}_{l_2, l_2}.$$

In addition, if  $k = \omega(1)$  and  $k = o(n)$ , we have

$$\lim_{n \rightarrow \infty} \frac{H(\mathbf{B}_{\text{ut}})}{\binom{n'}{2} H(\mathbf{B}_{12})} = 1. \quad (14)$$

**Length Analysis for Correlated Sequences** Thanks to this property of the block decomposition, we hope to compress these blocks as if they are independent using a Laplace probability assignment (which, recall, is universal for the class of all  $m$ -ary iid processes). However, since these blocks are still correlated (albeit weakly), we will need a result on the performance of Laplace probability assignment on correlated sequences with identical marginals, which we give next.

**Theorem 4** (Laplace probability assignment for correlated sequence). *Consider arbitrarily correlated  $Z_1, Z_2, \dots, Z_N$ , where the marginal distribution of each  $Z_i$  is identically distributed over an alphabet of size  $m \geq 2$ . Let  $\ell_{\text{L}}(z^N) = \log \frac{1}{q_{\text{L}}(z^N)}$  where  $q_{\text{L}}(\cdot)$  is the marginal distribution induced by Laplace probability assignment in (10)*

$$q_{\text{L}}(z^N) := \frac{N_1! N_2! \cdots N_m!}{N!} \cdot \frac{1}{\binom{N+m-1}{m-1}}. \quad (15)$$

We then have

$$\mathbb{E}[\ell_{\text{L}}(Z^N)] \leq m \log(2eN) + NH(Z_1). \quad (16)$$

We provide a similar result for the KT probability assignment.

**Theorem 5** (KT probability assignment for correlated sequence). *Consider arbitrarily correlated  $Z_1, Z_2, \dots, Z_N$ , where the marginal distribution of each  $Z_i$  is identically distributed over an alphabet of size  $m \geq 2$ . Let  $\ell_{\text{KT}}(z^N) = \log \frac{1}{q_{\text{KT}}(z^N)}$  where  $q_{\text{KT}}(\cdot)$  is the marginal distribution induced by KT probability assignment in (9)*

$$q_{\text{KT}}(z^N) = \frac{(2N_1 - 1)!! (2N_2 - 1)!! \cdots (2N_m - 1)!!}{m(m+2) \cdots (m+2N-2)} \quad (17)$$

with  $(-1)!! \triangleq 1$ . We then have

$$\mathbb{E}[\ell_{\text{KT}}(Z^N)] \leq \frac{m}{2} \log\left(e\left(1 + \frac{2N}{m}\right)\right) + \frac{1}{2} \log(\pi N) + NH(Z_1). \quad (18)$$

We are now ready to prove Theorem 1.

**Proof of Theorem 1.** We will prove the universality of  $C_k$  for both KT probability assignment and Laplace probability assignment. Note that the upper bound on the expected length of KT in (18) is upper bounded by the upper bound on the length of Laplace in (16). So it suffices to show Laplace probability assignment is universal.

We use the bound in Theorem 4 to establish the upper bound on the length of the code. Recall that here we compress the diagonal blocks  $\mathbf{B}_{\text{d}}$  ( $m = 2^{k_n^2}$ -sized alphabet,  $N = n'$  blocks) and the off-diagonal blocks  $\mathbf{B}_{\text{ut}}$  ( $m = 2^{k_n^2}$ -sized alphabet,  $N = \binom{n'}{2}$  blocks) separately. We have,

$$\begin{aligned} \frac{\mathbb{E}(\ell(C_k(A_n)))}{H(A_n)} &= \frac{\mathbb{E}(\ell_{\text{L}}(\mathbf{B}_{\text{ut}})) + \mathbb{E}(\ell_{\text{L}}(\mathbf{B}_{\text{d}}))}{H(A_n)} \\ &\leq \frac{\binom{n'}{2} H(\mathbf{B}_{12}) + 2^{k_n^2} \log\left(2e \binom{n'}{2}\right) + n' H(\mathbf{B}_{11}) + 2^{k_n^2} \log(2en')}{H(A_n)} \end{aligned}$$

$$\begin{aligned}
&\stackrel{(a)}{\leq} \frac{\binom{n'}{2}H(\mathbf{B}_{12}) + 2^{k_n^2} \log(en^2) + nH(\mathbf{B}_{11}) + 2^{k_n^2} \log(2en)}{H(A_n)} \\
&\stackrel{(b)}{\leq} \frac{\binom{n'}{2}H(\mathbf{B}_{12}) + 2^{k_n^2} \log(2e^2n^3) + nk_n^2H(A_{12})}{H(A_n)} \\
&= \frac{\binom{n'}{2}H(\mathbf{B}_{12})}{H(A_n)} + \frac{2^{k_n^2} \log(2e^2n^3)}{H(A_n)} + \frac{nk_n^2H(A_{12})}{H(A_n)},
\end{aligned}$$

where in (a) we bound  $\binom{n'}{2} \leq n^2$  and  $n' \leq n$ , and in (b) we note that  $H(\mathbf{B}_{11}) \leq k_n^2H(A_{12})$  since there are  $k_n^2 - k_n$  elements of the matrix (all apart from the diagonal elements) are distributed identically as  $A_{12}$ . We will now analyze each of these three terms separately. Firstly, using Theorem 3 yields that  $\frac{\binom{n'}{2}H(\mathbf{B}_{12})}{H(A_n)} \rightarrow 1$ . Next, since  $f(n) = \Omega\left(\frac{1}{n^{2-\epsilon}}\right)$ , we have  $H(A_n) = \Omega(n^\epsilon \log n)$  and subsequently substituting  $k_n \leq \sqrt{\delta \log n}$ , we have

$$\frac{2^{k_n^2} \log(2en^3)}{H(A_n)} = O\left(\frac{n^\delta \log n}{n^\epsilon \log n}\right) = O\left(n^{\delta-\epsilon}\right) = o(1)$$

since  $\delta < \epsilon$ . Moreover, we have

$$\frac{nk_n^2H(A_{12})}{H(A_n)} \leq \frac{nk_n^2H(A_{12})}{H(A_n|X^n)} = \frac{nk_n^2H(A_{12})}{\binom{n}{2}H(A_{12}|X_1, X_2)} = O\left(\frac{k_n^2}{n}\right) = o(1),$$

where the penultimate equality used the fact that  $H(A_{12}) \sim H(A_{12}|X_1, X_2)$  (since  $h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p}) \sim \mathbf{p}^T h(f(n)\mathbf{Q})\mathbf{p}$ ). We have then established that

$$\begin{aligned}
\frac{\mathbb{E}(\ell(C_k(A_n)))}{H(A_n)} &\leq \frac{\binom{n'}{2}H(\mathbf{B}_{12})}{H(A_n)} + \frac{2^{k_n^2} \log(2en^3)}{H(A_n)} + \frac{nk_n^2H(A_{12})}{H(A_n)} \\
&= 1 + o(1),
\end{aligned}$$

which finishes the proof.  $\square$

The proof of Theorem 2 follows similar arguments as in Theorem 1 and is deferred to Section 4.5.

## 4 Proof of Universality

### 4.1 Graph Entropy

*Proof of Lemma 1.* Note that

$$\begin{aligned}
H(A_n) &= H(A_n|X^n) + I(X^n; A_n) \\
&= \binom{n}{2}H(A_{12}|X_1, X_2) + I(X^n; A_n)
\end{aligned} \tag{19}$$

$$= \binom{n}{2}\mathbf{p}^T h(f(n)\mathbf{Q})\mathbf{p} + I(X^n; A_n), \tag{20}$$

where (20) follows since all the  $\binom{n}{2}$  edges are identically distributed and also independent given  $X^n$  and consequently

$$H(A_n|X^n) = \binom{n}{2} H(A_{12}|X_1, X_2) = \binom{n}{2} \sum_{i,j} H(A_{12}|X_1 = i, X_2 = j) p_i p_j = \binom{n}{2} \mathbf{p}^T h(f(n)\mathbf{Q}) \mathbf{p}.$$

When  $f(n) = \Theta(1)$ , we see that since

$$0 \leq I(X^n; A_n) \leq H(X^n) = nH(X_1) \leq n \log L,$$

we have that  $H(A_n) = \binom{n}{2} \mathbf{p}^T h(f(n)\mathbf{Q}) \mathbf{p} + o(n^2 h(f(n)))$ .

Next, consider the case when  $f(n) = o(1)$  and  $f(n) = \Omega\left(\frac{1}{n^2}\right)$ . By properties of the entropy, we have

$$H(A_n|X^n) \leq H(A_n) \leq \binom{n}{2} H(A_{12}). \quad (21)$$

Note that

$$P(A_{12} = 1) = \sum_{i,j} P(A_{12} = 1|X_1 = i, X_2 = j) p_i p_j = \mathbf{p}^T f(n)\mathbf{Q}\mathbf{p},$$

which yields that  $H(A_{12}) = h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})$ . Substituting this in (21) gives

$$\binom{n}{2} \mathbf{p}^T h(f(n)\mathbf{Q}) \mathbf{p} \leq H(A_n) \leq \binom{n}{2} h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p}). \quad (22)$$

Note now for any  $g(n) = o(1)$ , we have

$$\begin{aligned} h(g(n)) &= -g(n) \log g(n) - (1 - g(n)) \log(1 - g(n)) \\ &= -g(n) \log g(n) \left( 1 + \frac{(1 - g(n)) \log(1 - g(n))}{g(n) \log g(n)} \right). \end{aligned}$$

By noting that  $\frac{\log(1-g(n))}{g(n)} \rightarrow -1$  and  $\frac{1}{\log(g(n))} \rightarrow 0$  as  $g(n) \rightarrow 0$  we see that

$$h(g(n)) = g(n) \log \frac{1}{g(n)} (1 + o(1)).$$

Using this, we note that  $\mathbf{p}^T h(f(n)\mathbf{Q}) \mathbf{p} = \mathbf{p}^T \mathbf{Q} \mathbf{p} f(n) \log \frac{1}{f(n)} (1 + o(1))$  and  $h(f(n)\mathbf{p}^T \mathbf{Q} \mathbf{p}) = \mathbf{p}^T \mathbf{Q} \mathbf{p} f(n) \log \frac{1}{f(n)} (1 + o(1))$ . Finally, substituting this into (22) yields

$$H(A_n) = \binom{n}{2} \mathbf{p}^T \mathbf{Q} \mathbf{p} f(n) \log \frac{1}{f(n)} (1 + o(1))$$

as required. □

## 4.2 Asymptotic i.i.d. via Block Decomposition

We first invoke a known property of stochastic block models (see, for example, [35]). We include the proof here for completeness.

**Lemma 2** (Exchangeability of SBM). *Let  $A_n \sim \text{SBM}(n, L, \mathbf{p}, \mathbf{W})$ . For a permutation  $\pi : [n] \rightarrow [n]$ ,*

let  $\pi(A_n)$  be an  $n \times n$  matrix whose  $(i, j)$  entry is given by  $A_{\pi(i), \pi(j)}$ . Then, for any permutation  $\pi : [n] \rightarrow [n]$ , the joint distribution of  $A_n$  is the same as the joint distribution of  $\pi(A_n)$ , i.e.,

$$A_n \stackrel{d}{=} \pi(A_n). \quad (23)$$

*Proof.* Let  $a_n$  be a realization of the random matrix  $A_n$  and  $\pi(X^n)$  be the permuted vector  $(X_{\pi(1)}, \dots, X_{\pi(n)})$ . For any symmetric binary matrix  $a_n$  with zero diagonal entries, we have

$$\begin{aligned} \mathbb{P}(A_n = a_n) &= \sum_{x^n \in [L]^n} \mathbb{P}(A_n = a_n, X^n = x^n) \\ &= \sum_{x^n \in [L]^n} \mathbb{P}(A_n = a_n | X^n = x^n) \prod_{i=1}^n \mathbb{P}(X_i = x_i) \\ &\stackrel{(a)}{=} \sum_{x^n \in [L]^n} \prod_{\substack{i,j \\ 1 \leq i < j \leq n}} \mathbb{P}(A_{ij} = a_{ij} | X_i = x_i, X_j = x_j) \prod_{i=1}^n \mathbb{P}(X_{\pi(i)} = x_i) \\ &\stackrel{(b)}{=} \sum_{x^n \in [L]^n} \prod_{\substack{i,j \\ 1 \leq i < j \leq n}} (W_{x_i, x_j})^{a_{ij}} (1 - W_{x_i, x_j})^{1-a_{ij}} \prod_{i=1}^n \mathbb{P}(X_{\pi(i)} = x_i) \\ &\stackrel{(c)}{=} \sum_{x^n \in [L]^n} \prod_{\substack{i,j \\ 1 \leq i < j \leq n}} \mathbb{P}(A_{\pi(i), \pi(j)} = a_{ij} | X_{\pi(i)} = x_i, X_{\pi(j)} = x_j) \prod_{i=1}^n \mathbb{P}(X_{\pi(i)} = x_i) \\ &= \sum_{x^n \in [L]^n} \mathbb{P}(\pi(A_n) = a_n, \pi(X^n) = x^n) \\ &= \mathbb{P}(\pi(A_n) = a_n), \end{aligned}$$

where (a) follows since  $X^n$  are i.i.d. and thus  $\mathbb{P}(X_i = x_i) = \mathbb{P}(X_{\pi(i)} = x_i)$  and (b) follows since  $A_{ij} \sim \text{Bern}(W_{X_i, X_j})$ , and thus

$$\mathbb{P}(A_{ij} = a_{ij} | X_i = x_i, X_j = x_j) = \begin{cases} W_{x_i, x_j} & \text{if } a_{ij} = 1 \\ 1 - W_{x_i, x_j} & \text{if } a_{ij} = 0 \end{cases} \quad (24)$$

$$= (W_{x_i, x_j})^{a_{ij}} (1 - W_{x_i, x_j})^{1-a_{ij}}. \quad (25)$$

The step in (c) follows since  $A_{\pi(i), \pi(j)} \sim \text{Bern}(W_{X_{\pi(i)}, X_{\pi(j)}})$  and the conditional probability has the same expression as in (25).  $\square$

Now we are ready to establish Theorem 3.

*Proof of Theorem 3.* For any  $i_1 \neq j_1$  and  $i_2 \neq j_2$ , consider a permutation  $\pi_1 : [n] \rightarrow [n]$  that has

$$\pi_1(x) = \begin{cases} x + (i_2 - i_1)k_n & \text{for } (i_1 - 1)k_n + 1 \leq x \leq i_1 k_n \\ x + (j_2 - j_1)k_n & \text{for } (j_1 - 1)k_n + 1 \leq x \leq j_1 k_n \end{cases}$$

and the remaining  $n - 2k_n$  arguments are mapped to the  $n - 2k_n$  values in  $[n] \setminus \{(i_2 - 1)k_n + 1, \dots, i_2 k_n, (j_2 - 1)k_n, \dots, j_2 k_n\}$  in any order. Lemma 2 implies that  $\mathbf{B}_{i_1, j_1}$ , which is the submatrix formed by the rows  $(i_1 - 1)k_n + 1, \dots, i_1 k_n$  and the columns  $(j_1 - 1)k_n + 1, \dots, j_1 k_n$  has the same distribution as the submatrix formed by the rows  $\pi_1((i_1 - 1)k_n + 1), \dots, \pi_1(i_1 k_n)$  and the columns

$\pi_1((j_1 - 1)k_n + 1), \dots, \pi_1(j_1 k_n)$ . From the definition of  $\pi_1$ , we see that the latter submatrix is  $\mathbf{B}_{i_2, j_2}$  and we establish that  $\mathbf{B}_{i_1, j_1} \stackrel{d}{=} \mathbf{B}_{i_2, j_2}$ . Similarly, defining a permutation  $\pi_2 : [n] \rightarrow [n]$  which has

$$\pi_2(x) = x + (l_2 - l_1)k_n \quad \text{for } (l_1 - 1)k_n + 1 \leq x \leq l_1 k_n$$

and invoking Lemma 2 establishes  $\mathbf{B}_{l_1, l_1} \stackrel{d}{=} \mathbf{B}_{l_2, l_2}$ .

Now, clearly  $H(\mathbf{B}_{\text{ut}}) \leq \binom{n'}{2} H(\mathbf{B}_{12})$ , and therefore we have

$$\limsup_{n \rightarrow \infty} \frac{H(\mathbf{B}_{\text{ut}})}{\binom{n'}{2} H(\mathbf{B}_{12})} \leq 1. \quad (26)$$

Moreover we have  $H(A_n) = H(\mathbf{B}_{\text{ut}}, \mathbf{B}_{\text{d}}) \leq H(\mathbf{B}_{\text{ut}}) + H(\mathbf{B}_{\text{d}}) \leq H(\mathbf{B}_{\text{ut}}) + n' H(\mathbf{B}_{11}) \leq H(\mathbf{B}_{\text{ut}}) + n' k_n^2 h(A_{12})$  where the last inequality follows by noting that except for the diagonal elements of  $\mathbf{B}_{\text{d}}$  (which are zero and thus have zero entropy), all other elements have the same distribution as  $A_{12}$ . We therefore obtain  $H(\mathbf{B}_{\text{ut}}) \geq H(A_n) - n' k_n^2 h(A_{12}) = H(A_n) - n k_n h(A_{12}) \geq H(A_n | X_1^n) - n k_n h(A_{12}) = \binom{n}{2} \mathbf{p}^T h(f(n) \mathbf{Q}) \mathbf{p} - n k_n h(f(n) \mathbf{p}^T \mathbf{Q} \mathbf{p})$ . Consequently,

$$\frac{H(\mathbf{B}_{\text{ut}})}{\binom{n'}{2} H(\mathbf{B}_{12})} \geq \frac{\binom{n}{2} \left( \mathbf{p}^T h(f(n) \mathbf{Q}) \mathbf{p} - \frac{2k_n h(f(n) \mathbf{p}^T \mathbf{Q} \mathbf{p})}{n-1} \right)}{\binom{n'}{2} H(\mathbf{B}_{12})}. \quad (27)$$

We will now analyze the right hand side of (27) in two parameter ranges.

- $f(n) = 1$  : We have

$$\begin{aligned} H(\mathbf{B}_{12}) &\stackrel{(a)}{\leq} H(\mathbf{B}_{12} | X_1^{2k_n}) + H(X_1^{2k_n}) \\ &\leq H(\mathbf{B}_{12} | X_1^{2k_n}) + 2k_n H(\mathbf{p}) \\ &\stackrel{(b)}{=} k_n^2 H(A_{1, k_n} | X_1, X_{k_n}) + 2k_n H(\mathbf{p}) \\ &\leq k_n^2 \left( \mathbf{p}^T h(\mathbf{Q}) \mathbf{p} + 2 \frac{\log L}{k_n} \right), \end{aligned} \quad (28)$$

where (a) follows from the chain rule and (b) follows since all elements of the matrix  $\mathbf{B}_{12}$  are independent given  $X_1, \dots, X_{2k_n}$ . Plugging this into the right hand side of (27) we obtain

$$\frac{H(\mathbf{B}_{\text{ut}})}{\binom{n'}{2} H(\mathbf{B}_{12})} \geq \frac{\binom{n}{2} \left( \mathbf{p}^T h(\mathbf{Q}) \mathbf{p} - \frac{2k_n h(\mathbf{p}^T \mathbf{Q} \mathbf{p})}{n-1} \right)}{\binom{n'}{2} k_n^2 \left( \mathbf{p}^T h(\mathbf{Q}) \mathbf{p} + 2 \frac{\log L}{k_n} \right)}. \quad (29)$$

Since  $k_n = o(n)$ ,  $k_n = \omega(1)$  and  $\binom{n'}{2} k_n^2 \sim \binom{n}{2}$ , we have from (29)

$$\liminf_{n \rightarrow \infty} \frac{H(\mathbf{B}_{\text{ut}})}{\binom{n'}{2} H(\mathbf{B}_{12})} \geq 1, \quad (30)$$

which together with (26) yields the required result.

- $f(n) = \Omega\left(\frac{1}{n^2}\right)$ ,  $f(n) = o(1)$  : Since  $\mathbf{B}_{12}$  is a matrix of  $k_n^2$  identically distributed Bernoulli

random variables, we have

$$H(\mathbf{B}_{12}) \leq k_n^2 h(A_{1,k_n}) = k_n^2 h(f(n)\mathbf{p}^T \mathbf{Q} \mathbf{p}). \quad (31)$$

Plugging this into the RHS of (27) then yields

$$\frac{H(\mathbf{B}_{\text{ut}})}{\binom{n'}{2} H(\mathbf{B}_{12})} \geq \frac{\binom{n}{2} \left( \mathbf{p}^T h(f(n)\mathbf{Q}) \mathbf{p} - \frac{2k_n h(f(n)\mathbf{p}^T \mathbf{Q} \mathbf{p})}{n-1} \right)}{\binom{n'}{2} k_n^2 h(f(n)\mathbf{p}^T \mathbf{Q} \mathbf{p})}. \quad (32)$$

We first observe that in this parameter range, since  $f(n) = o(1)$ , we have by Lemma 1

$$\mathbf{p}^T h(f(n)\mathbf{Q}) \mathbf{p} \sim h(f(n)\mathbf{p}^T \mathbf{Q} \mathbf{p}). \quad (33)$$

Finally using that  $k_n = o(n)$  and  $\binom{n'}{2} k_n^2 \sim \binom{n}{2}$  establishes

$$\liminf_{n \rightarrow \infty} \frac{H(\mathbf{B}_{\text{ut}})}{\binom{n'}{2} H(\mathbf{B}_{12})} \geq 1, \quad (34)$$

which together with (26) yields the required result.  $\square$

### 4.3 Length of the Laplace Probability Assignment

*Proof of Theorem 4.* Let us first elaborate the relation between probability assignment and compression length. In Algorithm 1, the terms  $\log(q(x_{j+1}|x^j))$  are added up, which lead to the marginal probability implied by the sequential probability assignment

$$\sum_{j=0}^{N-1} \log(q(x_{j+1}|x^j)) = \log \left( \prod_{j=0}^{N-1} q(x_{j+1}|x^j) \right) = \log(q(x^N)). \quad (35)$$

The compression output length of Algorithm 1 is  $\lceil \log \frac{1}{q(x^N)} \rceil + 1$ .

Now we analyze the compression length of Laplace compressor for the sequence  $Z_1, Z_2, \dots, Z_N$ . Define  $\theta_i := \mathbb{P}(Z_1 = i)$ ,  $N_i := \sum_{k=1}^N \mathbb{1}\{Z_k = i\}$ ,  $i \in [m]$ . We have

$$\begin{aligned} \ell_L(z^n) &= \log \frac{1}{q_L(z^N)} \\ &= \log \frac{\theta_1^{N_1} \theta_2^{N_2} \dots \theta_m^{N_m}}{q_L(z^N)} + \log \frac{1}{\theta_1^{N_1} \theta_2^{N_2} \dots \theta_m^{N_m}} \\ &= \log \binom{N+m-1}{m-1} + \log \left( \frac{N!}{N_1! N_2! \dots N_m!} \theta_1^{N_1} \theta_2^{N_2} \dots \theta_m^{N_m} \right) + \log \frac{1}{\theta_1^{N_1} \theta_2^{N_2} \dots \theta_m^{N_m}} \\ &\stackrel{(a)}{\leq} \log \binom{N+m-1}{m-1} + \log \frac{1}{\theta_1^{N_1} \theta_2^{N_2} \dots \theta_m^{N_m}} \\ &\stackrel{(b)}{\leq} (m-1) \log \left( e \left( \frac{N}{m-1} + 1 \right) \right) + \log \frac{1}{\theta_1^{N_1} \theta_2^{N_2} \dots \theta_m^{N_m}} \end{aligned}$$

$$\leq m \log(2eN) + \sum_{i=1}^m N_i \log \frac{1}{\theta_i}, \quad (36)$$

where (a) follows since  $\frac{N!}{N_1!N_2!\dots N_m!}\theta_1^{N_1}\theta_2^{N_2}\dots\theta_m^{N_m}$  is a multinomial probability which is always upper bounded by 1, and (b) follows since  $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$ . Taking expectation on both sides of (36), we obtain

$$\begin{aligned} \mathbb{E}[\ell_L(Z^N)] &\leq m \log(2eN) + \sum_{i=1}^m \mathbb{E}[N_i] \log \frac{1}{\theta_i} \\ &\stackrel{(a)}{=} m \log(2eN) + \sum_{i=1}^m N\theta_i \log \frac{1}{\theta_i} \\ &= m \log(2eN) + NH(Z_1), \end{aligned}$$

where (a) follows since  $\mathbb{E}[N_i] = \sum_{k=1}^N \mathbb{E}[\mathbb{1}\{Z_k = i\}] = NP(Z_1 = i)$  since the  $Z_i$  are identically distributed.  $\square$

#### 4.4 Length of the KT probability assignment

**Lemma 3.** For any integer  $m > 0$ ,  $N_1, N_2, \dots, N_m \in \mathbb{N}$  and probability distribution  $(\theta_1, \dots, \theta_m)$ ,

$$\frac{\binom{N}{N_1, N_2, \dots, N_m} \theta_1^{N_1} \dots \theta_m^{N_m}}{\binom{2N}{2N_1, 2N_2, \dots, 2N_m} \theta_1^{2N_1} \dots \theta_m^{2N_m}} \geq 1,$$

where  $N = \sum_{i=1}^m N_i$ .

**Remark 4.** Equivalently, consider an urn containing known number of balls with  $m$  different colours. The lemma claims that the probability of getting  $N_1$  balls of colour 1,  $N_2$  of balls of colour 2,  $\dots$   $N_m$  balls of colour  $m$  out of  $N$  draws with replacement is always greater than the probability of getting  $2N_1$  balls of colour 1,  $2N_2$  of balls of colour 2,  $\dots$   $2N_m$  balls of colour  $m$  out of  $2N$  draws with replacement.

*Proof.* Let  $p_1 = N_1/N, p_2 = N_2/N, \dots, p_m = N_m/N$ . Notice that  $\sum_{i=1}^m p_i = 1$ , so  $(p_1, \dots, p_m)$  can be viewed as a probability distribution. And the entropy of this distribution is  $H(p_1, \dots, p_m) = \sum_{i=1}^m -p_i \log p_i$ . Firstly we consider the case when  $N_1, N_2, \dots, N_m$  are all positive and none of them equal to  $N$ . By Stirling's approximation for factorial  $\sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{1/(12n+1)} \leq n! \leq \sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{1/12n}$ , we can bound

$$\begin{aligned} \binom{N}{N_1, N_2, \dots, N_m} &\geq \frac{\sqrt{2\pi N} N^N \exp\left(\frac{1}{12N+1} - \frac{1}{12N_1} - \frac{1}{12N_2} - \dots - \frac{1}{12N_m}\right)}{(2\pi)^{m/2} (N_1 N_2 \dots N_m)^{1/2} N_1^{N_1} N_2^{N_2} \dots N_m^{N_m}} \\ &= \frac{\exp\left(\frac{1}{12N+1} - \frac{1}{12N_1} - \frac{1}{12N_2} - \dots - \frac{1}{12N_m}\right)}{(2\pi)^{\frac{m-1}{2}} (p_1 p_2 \dots p_m)^{1/2} N^{\frac{m-1}{2}} 2^{-NH(p_1, p_2, \dots, p_m)}}. \end{aligned}$$

Similarly, we have

$$\binom{2N}{2N_1, 2N_2, \dots, 2N_m} \leq \frac{\exp\left(\frac{1}{24N} - \frac{1}{24N_1+1} - \frac{1}{24N_2+1} - \dots - \frac{1}{24N_m+1}\right)}{(2\pi)^{\frac{m-1}{2}} 2^{\frac{m-1}{2}} (p_1 p_2 \dots p_m)^{1/2} N^{\frac{m-1}{2}} 2^{-2NH(p_1, \dots, p_m)}}.$$

Consider the function

$$f(N_1, N_2, \dots, N_m) = \frac{1}{12N+1} - \frac{1}{24N} + \left(\frac{1}{24N_1+1} - \frac{1}{12N_1}\right) + \left(\frac{1}{24N_2+1} - \frac{1}{12N_2}\right) + \dots + \left(\frac{1}{24N_m+1} - \frac{1}{12N_m}\right)$$

and the function

$$g(n) = \frac{1}{24n+1} - \frac{1}{12n},$$

where  $n$  is a positive integer. Function  $g(n)$  is minimized with  $n = 1$  and  $\min g(n) = 1/25 - 1/12$  and we can bound function  $f(N_1, N_2, \dots, N_m) \geq \frac{1}{12N+1} - \frac{1}{24N} + (1/25 - 1/12)m$ . Finally we are ready to prove the lemma.

$$\begin{aligned} \frac{\binom{N}{N_1, N_2, \dots, N_m} \theta_1^{N_1} \dots \theta_m^{N_m}}{\binom{2N}{2N_1, 2N_2, \dots, 2N_m} \theta_1^{2N_1} \dots \theta_m^{2N_m}} &\geq \frac{2^{\frac{m-1}{2}} \exp(f(N_1, N_2, \dots, N_m))}{2^{NH(p_1 \dots p_m)} \theta_1^{N_1} \dots \theta_m^{N_m}} \\ &\geq \frac{2^{\frac{m-1}{2}} \exp\left(\frac{1}{12N+1} - \frac{1}{24N} + (1/25 - 1/12)m\right)}{2^{-ND_{\text{KL}}(p||\theta)}} \\ &= 2^{\frac{m-1}{2}} 2^{ND_{\text{KL}}(p||\theta)} 2^{\log e\left(\frac{1}{12N+1} - \frac{1}{24N} + (1/25 - 1/12)m\right)}. \end{aligned}$$

Notice that  $\frac{1}{12N+1} - \frac{1}{24N}$  goes to zero when  $N \rightarrow \infty$ ,  $\frac{m-1}{2} > (1/25 - 1/12)m$  and  $D_{\text{KL}}(P||\theta) \geq 0$ . Therefore in this case,

$$\frac{\binom{N}{N_1, N_2, \dots, N_m} \theta_1^{N_1} \dots \theta_m^{N_m}}{\binom{2N}{2N_1, 2N_2, \dots, 2N_m} \theta_1^{2N_1} \dots \theta_m^{2N_m}} \geq 1.$$

When one of  $\{N_i\}_{i=1}^N$  equals to  $N$ , without loss of generality, we assume that  $N_1 = N$ . We have

$$\frac{\binom{N}{N_1, N_2, \dots, N_m} \theta_1^{N_1} \dots \theta_m^{N_m}}{\binom{2N}{2N_1, 2N_2, \dots, 2N_m} \theta_1^{2N_1} \dots \theta_m^{2N_m}} = \frac{1}{\theta_1^{N_1} \dots \theta_m^{N_m}} > 1.$$

When there are  $k$  numbers out of  $N_1, N_2, \dots, N_m$  that equal to zero, we can simply remove these values and consider the case with alphabet size  $m - k$ . And this will yield the same result.  $\square$

*Proof of Theorem 5.* In this proof, we define a generalized form of factorial function. Let  $x$  be a positive integer,  $(x + \frac{1}{2})! = \frac{1}{2} \cdot \frac{3}{2} \cdots (x + \frac{1}{2})$ . Since  $(2N_1 - 1)!! = \frac{(2N_1)!}{2^{N_1}(N_1)!}$ , we have

$$m(m+2) \cdots (m+2N-2) = 2^N \binom{m}{2} \binom{m+2}{2} \cdots \binom{m+2N-2}{2} = 2^N \frac{(\frac{m}{2} + N - 1)!}{(\frac{m}{2} - 1)!}.$$

Therefore we can rewrite the KT probability assignment in (17) as

$$\begin{aligned} q_{\text{KT}}(z^N) &= \frac{(\frac{m}{2} - 1)!}{2^N (\frac{m}{2} + N - 1)!} \frac{\binom{2N}{N}}{\binom{2N}{N}} \prod_{i=1}^m \frac{(2N_i)!}{N_i! 2^{N_i}} \\ &= \frac{(\frac{m}{2} - 1)!}{2^N (\frac{m}{2} + N - 1)!} \binom{2N}{N} N! \frac{N!}{(2N)!} \prod_{i=1}^m \frac{(2N_i)!}{N_i! 2^{N_i}} \\ &\stackrel{(a)}{\geq} \frac{(\frac{m}{2} - 1)! \binom{2N}{N}}{4^N (N + \frac{m}{2} - \frac{1}{2})^{\frac{m-1}{2}}} \frac{N!}{(2N)!} \prod_{i=1}^m \frac{(2N_i)!}{N_i!} \end{aligned}$$

$$\stackrel{(b)}{=} \frac{\theta_1^{N_1} \dots \theta_m^{N_m} (\frac{m}{2} - 1)! \binom{2N}{N}}{4^N (N + \frac{m}{2} - \frac{1}{2})^{\frac{m-1}{2}}} \frac{\binom{N}{N_1, N_2, \dots, N_m} \theta_1^{N_1} \dots \theta_m^{N_m}}{\binom{2N}{2N_1, 2N_2, \dots, 2N_m} \theta_1^{2N_1} \dots \theta_m^{2N_m}},$$

where (a) follows that when  $m$  is even,  $\frac{N!}{(\frac{m}{2} + N - 1)!} = \frac{1}{(N+1) \dots (\frac{m}{2} + N - 1)} \geq \frac{1}{(N + \frac{m}{2} - \frac{1}{2})^{\frac{m-1}{2}}}$  and when  $m$  is odd,  $\frac{N!}{(\frac{m}{2} + N - 1)!} \geq \frac{N!}{(\frac{m}{2} + N - \frac{1}{2})!} = \frac{1}{(N+1) \dots (\frac{m}{2} + N - \frac{1}{2})} \geq \frac{1}{(N + \frac{m}{2} - \frac{1}{2})^{\frac{m-1}{2}}}$ , (b) follows that  $\binom{N}{N_1, N_2, \dots, N_m} = \frac{N!}{\prod_{i=1}^m N_i!}$  and  $\theta_i \triangleq \mathbb{P}(Z_1 = i)$ . By lemma 3, we have  $q_{\text{KT}}(z^N) \geq \frac{\theta_1^{N_1} \dots \theta_m^{N_m} (\frac{m}{2} - 1)! \binom{2N}{N}}{4^N (N + \frac{m}{2} - \frac{1}{2})^{\frac{m-1}{2}}}$ . Thus,

$$\begin{aligned} \ell_{\text{KT}}(z^N) &= \log \frac{1}{q_{\text{KT}}(z^N)} \\ &\leq \log \frac{1}{\theta_1^{N_1} \dots \theta_m^{N_m}} + \log \frac{4^N (N + \frac{m}{2} - \frac{1}{2})^{\frac{m-1}{2}}}{(\frac{m}{2} - 1)! \binom{2N}{N}} \\ &= \log \frac{1}{\theta_1^{N_1} \dots \theta_m^{N_m}} + \frac{m-1}{2} \log \left( N + \frac{m-1}{2} \right) + \log \frac{4^N}{\binom{2N}{N}} - \log \left( \frac{m}{2} - 1 \right)! \\ &\stackrel{(a)}{\leq} \log \frac{1}{\theta_1^{N_1} \dots \theta_m^{N_m}} + \frac{m-1}{2} \log \left( N + \frac{m-1}{2} \right) + \log \frac{4^N}{\binom{2N}{N}} - \left( \frac{m}{2} - 1 \right) \log \left( \frac{\frac{m}{2} - 1}{e} \right) \\ &\stackrel{(b)}{\approx} \log \frac{1}{\theta_1^{N_1} \dots \theta_m^{N_m}} + \frac{m-1}{2} \log \left( N + \frac{m-1}{2} \right) + \log \sqrt{\pi N} - \left( \frac{m}{2} - 1 \right) \log \left( \frac{\frac{m}{2} - 1}{e} \right) \\ &\sim \frac{m}{2} \log \frac{e(\frac{m}{2} + N)}{m/2} + \log \sqrt{\pi N} + \log \frac{1}{\theta_1^{N_1} \dots \theta_m^{N_m}} \\ &= \frac{m}{2} \log \left( e \left( 1 + \frac{2N}{m} \right) \right) + \frac{1}{2} \log(\pi N) + \sum_{i=1}^m N_i \log \frac{1}{\theta_i}, \end{aligned}$$

where (a) follows Stirling's approximation  $k! \geq \sqrt{2\pi k} \left(\frac{k}{e}\right)^k e^{\frac{1}{12k+1}}$  and (b) follows Stirling's approximation for binomial coefficient, i.e.,  $\binom{2N}{N} \sim \frac{4^N}{\sqrt{\pi N}}$ . Therefore, we have

$$\mathbb{E}[\ell_{\text{KT}}(Z^N)] \leq \frac{1}{2} m \log \left( e \left( 1 + \frac{2N}{m} \right) \right) + \frac{1}{2} \log(\pi N) + NH(Z_1). \quad \square$$

## 4.5 Proof of Theorem 2

*Proof.* Once again, we establish universality for both KT and Laplace probability assignment. Following a similar argument as in the proof of Theorem 1, it suffices to show the universality of Laplace. Since we are compressing  $N = \binom{n}{2}$  identically distributed bits using a Laplace probability assignment, Theorem 4 yields

$$\begin{aligned} \frac{\mathbb{E}(\ell(C_1(A_n)))}{H(A_n)} &\leq \frac{\log(2eN) + NH(A_{12})}{H(A_n)} \\ &\leq \frac{\log(2eN) + NH(A_{12})}{H(A_n | X_1^n)} \\ &= \left( \frac{\log(2eN) + NH(A_{12})}{NH(A_{12})} \right) \frac{H(A_{12})}{H(A_{12} | X_1, X_2)} \end{aligned}$$

$$\begin{aligned}
&= \left( 1 + \frac{\log(2eN)}{Nh(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})} \right) \frac{h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})}{\mathbf{p}^T h(f(n)\mathbf{Q})\mathbf{p}} \\
&\stackrel{(a)}{=} 1 + o(1).
\end{aligned}$$

Here, (a) is justified by noting that  $\frac{\log(2eN)}{Nh(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})} \leq \frac{\log(2en^2)}{\binom{n}{2}h(n^{-(2-\epsilon)}\mathbf{p}^T\mathbf{Q}\mathbf{p})} \frac{h(n^{-(2-\epsilon)}\mathbf{p}^T\mathbf{Q}\mathbf{p})}{h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})}$ , and then noting that  $\frac{\log(2en^2)}{\binom{n}{2}h(n^{-(2-\epsilon)}\mathbf{p}^T\mathbf{Q}\mathbf{p})} = o(1)$  and  $\frac{h(n^{-(2-\epsilon)}\mathbf{p}^T\mathbf{Q}\mathbf{p})}{h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})} = O(1)$  when  $f(n) = \Omega\left(\frac{1}{n^{2-\epsilon}}\right)$  and that  $H(h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})) \sim \mathbf{p}^T h(f(n)\mathbf{Q})\mathbf{p}$ .  $\square$

**Remark 5.** When  $f(n) = 1$ , the compressor  $C_1$  is strictly suboptimal. This is because the length achieved by  $C_1$  is  $\binom{n}{2}h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})(1 + o(1))$ , whereas the first order term in the entropy is  $\binom{n}{2}\mathbf{p}^T h(f(n)\mathbf{Q})\mathbf{p}$ . When  $f(n)$  is  $o(1)$ , these two have the same first order term. However, when  $f(n)$  is constant,  $\mathbf{p}^T h(f(n)\mathbf{Q})\mathbf{p}$  is strictly smaller than  $h(f(n)\mathbf{p}^T\mathbf{Q}\mathbf{p})$  by concavity of entropy.

## 5 Second order analysis in the sparse regime

So far, we have shown that our algorithm always matches the first order term in the Shannon entropy. Now, we proceed to analyze the second order term of the expected length of our proposed compressor. We focus on the family of *symmetric* SBM with edge probability  $f(n) = 1/n$  and evaluate the performance of our compressor using the framework of local weak convergence, as introduced in [27]. This would allow us to compare the performance of our compressor to the compressor proposed in [14]. We first introduce some basic definitions on rooted graphs in Subsection 5.1. Then, we define the local weak convergence of graphs and derive the local weak convergence limit of the symmetric stochastic block model in Subsection 5.2. Finally, we review the definition of BC entropy in Subsection 5.3 and state the performance guarantee of our compression algorithm in Subsection 5.4.

### 5.1 Basic definitions on rooted graphs

Let  $G = (V, E)$  be a simple graph (undirected, unweighted, no self-loop), with  $V$  a countable set of vertices and  $E$  a countable set of edges. Let  $u \stackrel{G}{\sim} v$  denote the connectivity of vertices  $u$  and  $v$  in  $G$ .  $G$  is said to be *locally finite* if, for all  $v \in V$ , the degree of  $v$  in  $G$  is finite. A rooted graph  $(G, o)$  is a locally finite and connected graph  $G = (V, E, o)$  with a distinguished vertex  $o \in V$ , called the root. Two rooted graphs  $(G_1, o_1) = (V_1, E_1, o_1)$  and  $(G_2, o_2) = (V_2, E_2, o_2)$  are *isomorphic*, denoted as  $(G_1, o_1) \simeq (G_2, o_2)$ , if there exists a bijection  $\pi : V_1 \rightarrow V_2$  such that  $\pi(o_1) = o_2$  and  $u \stackrel{G_1}{\sim} v$  if and only if  $\pi(u) \stackrel{G_2}{\sim} \pi(v)$  for all  $u, v \in V_1$ . One can verify that this notion of isomorphism defines an equivalence relation on rooted graphs. Let  $[G, o]$  denote the equivalence class corresponding to  $(G, o)$ . Let  $\mathcal{G}^*$  denote the set of all locally finite and connected rooted graphs. For  $(G, o) \in \mathcal{G}^*$  and  $h \in \mathbb{N}$ , we write  $(G, o)_h$  for the truncated graph at depth  $h$  of the graph  $(G, o)$ , in other words, the induced subgraph on the vertices such that their distance from the root is less than or equal to  $h$ . The equivalence classes  $[G, o]_h$  follows the similar definition. Let  $\mathcal{G}_h^*$  denote the set of all  $[G, o]_h$ . Now, we define the metric  $d^*$  on  $\mathcal{G}^*$ . For any  $[G_1, o_1]$  and  $[G_2, o_2]$ , let

$$\hat{h} := \sup\{h \in \mathbb{Z}^+ : (G_1, o_1)_h \simeq (G_2, o_2)_h \text{ for some } (G_1, o_1) \in [G_1, o_1], (G_2, o_2) \in [G_2, o_2]\}$$

and define the metric  $d^*$  as

$$d^*([G_1, o_1], [G_2, o_2]) := \frac{1}{1 + \hat{h}}.$$

As shown in [14], equipped with the metric defined above,  $\mathcal{G}^*$  is a Polish space, i.e, a complete separable metric space. For this Polish space, let  $\mathcal{P}(\mathcal{G}^*)$  denote the Borel probability measures on it. We say that a sequence of measures  $\mu_n \in \mathcal{P}(\mathcal{G}^*)$  converges weakly to  $\mu \in \mathcal{P}(\mathcal{G}^*)$ , written as  $\mu_n \rightsquigarrow \mu$ , if for any bounded continuous function  $f$  on  $\mathcal{G}^*$ , we have  $\int f d\mu_n \rightarrow \int f d\mu$ . It was shown in [36] that  $\mu_n \rightsquigarrow \mu$  if for any uniformly continuous and bounded functions  $f$ , we have  $\int f d\mu_n \rightarrow \int f d\mu$ . For  $\mu \in \mathcal{P}(\mathcal{G}^*)$ ,  $h \in \{0, 1, 2, \dots\}$ , and  $[G, o] \in \mathcal{G}^*$ , let  $\mu_h$  denote the  $h$ -neighborhood marginal of  $\mu$

$$\mu_h([G, o]) = \sum_{[G', o] \in \mathcal{G}^*: [G', o]_h = [G, o]} \mu([G', o]).$$

For a locally finite graph  $G = (V, E)$  and a vertex  $v \in V$ , let  $G(v)$  denote the graph component in  $G$  that is connected to  $v$ . By our previous definitions,  $(G(v), v)$  denotes the rooted graph of the connected component of  $v$  and the root is located at  $v$  and  $[G(v), v]$  denotes the equivalence class corresponding to  $(G(v), v)$ . Now, the *rooted neighbourhood distribution* of  $G$  is defined as the distribution of the rooted graph when the root is chosen uniformly at random over  $V$

$$U(G) := \frac{1}{|V|} \sum_{v \in V} \delta_{[G(v), v]}, \quad (37)$$

where  $\delta$  is the Dirac delta function.

## 5.2 Local weak convergence

For our study of stochastic block model, which is a sequence of *random* graphs  $\{A_n\}_{n=1}^\infty$ ,  $U(A_n)$  as defined in (37) becomes a random distribution. In the section, we establish the asymptotic behavior of the average neighbourhood distribution  $EU(A_n)$  averaged over the randomness of the graph  $A_n$ .

To state the limiting distribution, we define the *Galton–Watson tree* probability distribution on rooted trees  $\text{GWT}(\text{P}_\lambda)$  as follows. Let  $\text{P}_\lambda$  denote the Poisson distribution with mean  $\lambda$ . We take a vertex as the root and generate  $Z^{(1)} \sim \text{P}_\lambda$  as the number of children of the first generation. For the first generation, independent of  $Z^{(1)}$ , we generate  $\xi_1^{(1)}, \dots, \xi_{Z^{(1)}}^{(1)}$  i.i.d. according to  $\text{P}_\lambda$  as the number of children of each vertex in the first generation. Let  $Z^{(2)} = \sum_{i=1}^{Z^{(1)}} \xi_i^{(1)}$  denote the total number of vertices in the first generation. In general, for the  $j$ th generation,  $j = 1, 2, \dots$ , generate the number of children for each vertex in the  $j$ th generation  $\xi_1^{(j)}, \dots, \xi_{Z^{(j)}}^{(j)}$  i.i.d. according to  $\text{P}_\lambda$ , independent of all previous variables  $\{\xi_1^{(i-1)}, \dots, \xi_{Z^{(i-1)}}^{(i-1)}, Z^{(i)}, \text{ for all } i \leq j\}$ . Let  $Z^{(j+1)} = \sum_{k=1}^{Z^{(j)}} \xi_k^{(j)}$  denote the total number of vertices in the  $j$ th generation. In this way, we iteratively defined a measure on rooted trees. With the definitions above, we are ready to establish the local weak convergence of the symmetric stochastic block model.

**Lemma 4** (Local weak convergence of sparse symmetric SBMs). *Let  $A_n$  denote a graph generated from a symmetric stochastic block model  $\text{SBM}(n, L, \mathbf{p}, \frac{1}{n}\mathbf{Q})$  with  $\mathbf{p} = (\frac{1}{L}, \dots, \frac{1}{L})$ ,  $\mathbf{Q}_{ii} = a, \forall i \in [n]$  and  $\mathbf{Q}_{ij} = b, \forall i, j \in [n], i \neq j$ . Let  $U(A_n)$ , defined as in (37), be the random rooted neighbourhood distribution of  $A_n$ . Then, the average neighbourhood distribution  $EU(A_n)$  converges weakly to a Poisson Galton–Walson tree*

$$EU(A_n) \rightsquigarrow \text{GWT}(\text{P}_\lambda),$$

where  $\lambda = \frac{a+(L-1)b}{L}$ .

**Remark 6.** When  $a = b$ , the symmetric stochastic block model recovers the well-known local weak convergence result on Erdős–Rényi model (see, e.g., []).

**Proof of Lemma 4.** We want to show that for any uniformly continuous and bounded function  $f$ ,

$$\left| \int f dEU(A_n) - \int f dGWT(P_\lambda) \right| \rightarrow 0$$

as  $n \rightarrow \infty$ . Since  $f$  is a uniformly continuous function on  $\mathcal{G}^*$ , for every  $\epsilon > 0$  there exists  $\delta > 0$  such that, for any pair of rooted graphs  $[G_1, o_1]$  and  $[G_2, o_2] \in \mathcal{G}^*$  with  $d^*([G_1, o_1], [G_2, o_2]) < \delta$  we have  $|f(G_1, o_1) - f(G_2, o_2)| < \epsilon$ . Recall that  $d^*([G_1, o_1], [G_2, o_2]) := \frac{1}{1+\hat{h}}$ , where  $\hat{h}$  denotes the maximum layers of matching between  $[G_1, o_1]$  and  $[G_2, o_2]$ . Therefore, as long as  $h > \frac{1}{\delta} - 1$ , we have  $|f((G, o)_h) - f(G, o)| < \epsilon$ . It follows that  $|f([i, o]) - f([g, o])| < \epsilon$ , if  $[i, o]_h = [g, o]$ . Let  $\mu \in \mathcal{P}(\mathcal{G}^*)$  and assume  $h > \frac{1}{\delta} - 1$ . We have

$$\left| \int f d\mu_h - \int f d\mu \right| = \left| \sum_{[g, o] \in \mathcal{G}_h^*} f([g, o])\mu_h([g, o]) - \sum_{[i, o] \in \mathcal{G}^*} f([i, o])\mu([i, o]) \right| \quad (38)$$

$$\leq \sum_{[g, o] \in \mathcal{G}_h^*} \left| f([g, o])\mu_h([g, o]) - \sum_{[i, o] \in \mathcal{G}^*: [i, o]_h = [g, o]} f([i, o])\mu([i, o]) \right| \quad (39)$$

$$= \sum_{[g, o] \in \mathcal{G}_h^*} \left| \sum_{[i, o] \in \mathcal{G}^*: [i, o]_h = [g, o]} (f([g, o]) - f([i, o]))\mu([i, o]) \right| \quad (40)$$

$$\leq \sum_{[g, o] \in \mathcal{G}_h^*} \sum_{[i, o] \in \mathcal{G}^*: [i, o]_h = [g, o]} |f([g, o]) - f([i, o])| \mu([i, o]) \quad (41)$$

$$\leq \sum_{[g, o] \in \mathcal{G}_h^*} \sum_{[i, o] \in \mathcal{G}^*: [i, o]_h = [g, o]} \epsilon \mu([i, o]) = \epsilon, \quad (42)$$

where (3) follows since  $\mu_h([g, o]) = \sum_{[i, o] \in \mathcal{G}^*: [i, o]_h = [g, o]} \mu([i, o])$ . Therefore, we have  $|\int f dEU(A_n)_h - \int f dEU(A_n)| < \epsilon$  and  $|\int f dGWT(P_\lambda)_h - \int f dGWT(P_\lambda)| < \epsilon$ . Let  $B \subseteq \mathcal{G}^*$  be a measurable event in  $\mathcal{G}^*$ . By the exchangeability of stochastic block model, we have  $EU(A_n)(B) = \frac{1}{n} \sum_{i=1}^n \mathbf{P}([A_n(i), i] \in B) = \mathbf{P}([A_n(1), 1] \in B)$ , in other words,  $EU(A_n)$  is simply the neighbourhood distribution at vertex 1. By the analogous argument as in proposition 2 of [34], for any  $\epsilon > 0$ , there exists  $n_0$  such that if  $n \geq n_0$  and  $\frac{\ln n}{10 \ln(2(a+(L-1)b))} \geq R$ , we have  $d_{TV}(GWT(P_\lambda)_R, EU(A_n)_R) < \epsilon$ , where  $d_{TV}(\cdot, \cdot)$  denotes the total variation distance between two measures. Remember here the total variation distance is  $d_{TV}(\mu_1, \mu_2) := \sup_{g: \mathcal{G}^* \rightarrow [-1, 1]} (\int g d\mu_1 - \int g d\mu_2)$ . Since  $f$  is a bounded function, we have  $|\int f dGWT(P_\lambda)_R - \int f dEU(A_n)_R| < \epsilon$ , as long as  $n$  is large enough. Therefore, if we take  $n$  large enough such that  $\frac{\ln n}{10 \ln(2(a+(L-1)b))} > \frac{1}{\delta} - 1$  and  $|\int f dGWT(P_\lambda)_h - \int f dEU(A_n)_h| < \epsilon$ , we have

$$\begin{aligned} \left| \int f dEU(A_n) - \int f dGWT(P_\lambda) \right| &\leq \left| \int f dEU(A_n)_h - \int f dEU(A_n) \right| \\ &\quad + \left| \int f dGWT(P_\lambda)_h - \int f dGWT(P_\lambda) \right| \\ &\quad + \left| \int f dGWT(P_\lambda)_h - \int f dEU(A_n)_h \right| \\ &< 3\epsilon, \end{aligned}$$

which completes the proof.  $\square$

### 5.3 BC entropy

In this section, we review the notion of BC entropy introduced in [27], which is shown to be the fundamental limit of universal lossless compression for certain graph family [14].

For a Polish space  $\Omega$ , let  $\mathcal{P}(\Omega)$  denote the set of all Borel probability measures on  $\Omega$ . Let  $A$  be a Borel set in  $\Omega$ , we define the  $\epsilon$ -extension of  $A$ , denoted  $A^\epsilon$ , as the union of the open balls with radius  $\epsilon$  centered around the points in  $A$ . For two probability measures  $\mu$  and  $\nu$  in  $\mathcal{P}(\Omega)$ , we define the Lévy–Prokhorov distance  $d_{\text{LP}}(\mu, \nu) := \inf\{\epsilon > 0 : \mu(A) \leq \nu(A^\epsilon) + \epsilon \text{ and } \nu(A) \leq \mu(A^\epsilon) + \epsilon, \forall A \in \mathcal{B}(\Omega)\}$ , where  $\mathcal{B}(\Omega)$  denotes the Borel sigma algebra of  $\Omega$ . Let  $\rho \in \mathcal{P}(\mathcal{G}^*)$ . Let  $d$  be the expected number of neighbours of root under the law  $\rho$  and let a sequence  $m = m(n)$  such that  $m/n \rightarrow d/2$ , as  $n \rightarrow \infty$ . Define  $\mathcal{G}_{n,m}$  to be the set of graphs with  $n$  vertices and  $m$  edges. For  $\epsilon > 0$ , define

$$\mathcal{G}_{n,m}(\rho, \epsilon) = \{G \in \mathcal{G}_{n,m} : U(G) \in B(\rho, \epsilon)\},$$

where  $B(\rho, \epsilon)$  denotes the open ball with radius  $\epsilon$  around  $\rho$  with respect to Lévy–Prokhorov metric. Now, we define the  $\epsilon$ -upper BC entropy of  $\rho$  as

$$\bar{\Sigma}(\rho, \epsilon) = \limsup_{n \rightarrow \infty} \frac{\log |\mathcal{G}_{n,m}(\rho, \epsilon)| - m \log n}{n}$$

and define the upper BC entropy of  $\rho$  as

$$\bar{\Sigma}(\rho) = \lim_{\epsilon \rightarrow 0} \bar{\Sigma}(\rho, \epsilon).$$

Similarly we define the  $\epsilon$ -lower BC entropy  $\underline{\Sigma}(\rho, \epsilon)$  and lower BC entropy  $\underline{\Sigma}(\rho)$  with lim sup replaced by lim inf in above definitions. If  $\rho$  is such that  $\bar{\Sigma}(\rho) = \underline{\Sigma}(\rho)$ , then this common limit is called the BC entropy of  $\rho$

$$\Sigma(\rho) := \bar{\Sigma}(\rho) = \underline{\Sigma}(\rho).$$

The following lemma states the BC entropy of the Galton–Watson tree distribution.

**Lemma 5** (Corollary 1.4 of [27]). *The BC entropy of the Galton–Watson tree distribution  $\text{GWT}(\mathbb{P}_\lambda)$  is given by*

$$\Sigma(\text{GWT}(\mathbb{P}_\lambda)) = \frac{\lambda}{2} \log \frac{e}{\lambda} \quad \text{bits.}$$

### 5.4 Achieving BC entropy in the sparse regime

With the Lemma above, we can give a performance guarantee of our algorithm corresponding to the BC entropy. It is a Theorem analog to Proposition 1 in [14].

**Theorem 6.** *Let  $A_n \sim \text{SBM}(n, L, \mathbf{p}, \frac{1}{n}\mathbf{Q})$  with  $\mathbf{p} = (\frac{1}{L}, \dots, \frac{1}{L})$ ,  $\mathbf{Q}_{ii} = a, \forall i \in [n]$  and  $\mathbf{Q}_{ij} = b, \forall i, j \in [n], i \neq j$ . Let  $\lambda = \mathbf{p}^T \mathbf{Q} \mathbf{p} = \frac{a+(L-1)b}{L}$  and  $m = \binom{n}{2} \frac{\lambda}{n}$  be the expected number of edges in the model. Then, our compression algorithm achieves the BC entropy of the local weak limit of stochastic block models in the sense that*

$$\limsup_{n \rightarrow \infty} \frac{\mathbb{E}[\ell(C_k(A_n))] - m \log n}{n} \leq \Sigma(\text{GWT}(\mathbb{P}_\lambda)).$$

*Proof.* By our proof of theorem (need to fill in ref), we have

$$\mathbb{E}[\ell(C_k(A_n))] \leq \binom{n'}{2} H(\mathbf{B}_{12}) + 2^{k^2} \log(2en^3) + nk_n^2 H(A_{12}).$$

Notice that

$$\begin{aligned}
\binom{n'}{2} H(\mathbf{B}_{12}) &\leq \binom{n'}{2} k_n^2 H(A_{12}) \\
&= \binom{n'}{2} k_n^2 h(\lambda/n) \\
&\stackrel{(1)}{=} \binom{n'}{2} k_n^2 \left( \frac{1}{n} \lambda \log \frac{ne}{\lambda} + o\left(\frac{1}{n}\right) \right) \\
&\stackrel{(2)}{\sim} \binom{n}{2} \left( \frac{1}{n} \lambda \log n + \frac{1}{n} \lambda \log \frac{e}{\lambda} + o\left(\frac{1}{n}\right) \right) \\
&= \binom{n}{2} \frac{1}{n} \lambda \log n + \frac{\lambda \log e - \lambda \log \lambda}{2} n + o(n) \\
&\stackrel{(3)}{=} m \log n + n \Sigma(\text{GWT}(\mathbb{P}_\lambda)) + o(n)
\end{aligned}$$

where (1) follows since  $h(p) = p \log \frac{e}{p} - \frac{\log e}{2} p^2 + o(p^2)$ , (2) follows since  $n'k_n = n$  and (3) follows from Lemma 5. Then it suffices to that the remaining terms in the upper bound of  $\mathbb{E}[\ell(C_k(A_n))]$  are all  $o(n)$ . Indeed we have

$$2^{k_n^2} \log(2en^3) \leq 2^{\delta \log n} \log(2en^3) = n^\delta \log(2en^3) = o(n)$$

since  $\delta < 1$  and

$$\begin{aligned}
nk_n^2 H(A_{12}) &= nk_n^2 h(\lambda/n) \\
&= nk_n^2 \left( \frac{1}{n} \lambda \log \frac{ne}{\lambda} + o\left(\frac{1}{n}\right) \right) \\
&\leq n\delta \log n \left( \frac{1}{n} \lambda \log \frac{ne}{\lambda} + o\left(\frac{1}{n}\right) \right) \\
&= \delta \log n \left( \lambda \log \frac{ne}{\lambda} \right) + o(\log n) \\
&= o(n).
\end{aligned}$$

□

**Remark 7.** For sparse symmetric SBMs, Theorem 6 shows that our compressor achieves the BC entropy of the Galton–Watson tree that is the local weak convergence limit of the underlying sequence of graphs. We note, however, that for the family of sparse symmetric SBMs, it is unclear if this BC entropy is the fundamental limit of lossless compression. This is because the family of sparse symmetric SBMs does not belong to the family of random graphs considered in [26], where a converse statement can be established.

## 6 Stationarity in the stochastic block model

In this section, we take a closer look at the correlation among entries in the adjacency matrix and explain why existing universal compressors developed for stationary processes may not be immediately applicable for certain orderings of the entries.

Compressing  $A_n$  entails compressing

$$A_{12}, \dots, A_{1,n}, A_{23}, \dots, A_{n-1,n},$$

i.e. the bits in the upper triangle of  $A_n$ . Clearly, these are not independent (because of the dependency through  $X_1^n$ ) so one cannot use any of the compressors universal for the class of iid processes to compress  $A_n$ . So, one hopes that it is possible to list the  $\binom{n}{2}$  random variables  $A_{12}, \dots, A_{1,n}, A_{23}, \dots, A_{n-1,n}$  in an order that makes the resulting sequence stationary, so that the Lempel–Ziv compressor (which, recall, is universal for the class of stationary processes) may be used. However, we show now that some of the most natural orders of listing these  $\binom{n}{2}$  bits result in a sequence that is nonstationary.

1. **Horizontally:** Listing the bits in the upper triangle row-wise (i.e. first listing the bits in the first row, followed by the bits in the second and so on, ending with  $A_{n-1,n}$ ) we get the following sequence

$$A_{12}, \dots, A_{1,n}, A_{23}, \dots, A_{2,n}, \dots, A_{n-1,n},$$

which can be seen to be nonstationary. Consider the case when  $n = 4, L = 2, Q_{11} = Q_{12} = 1, Q_{12} = 0$ . In this case the horizontal ordering is

$$A_{12}, A_{13}, A_{14}, A_{23}, A_{24}, A_{34}$$

and this is seen to be nonstationary by observing  $P(A_{12} = 1, A_{13} = 0, A_{14} = 1) > 0$  but  $P(A_{23} = 1, A_{24} = 0, A_{34} = 1) = 0$ .

2. **Vertically:** Listing the bits in the upper triangle column-wise (i.e. first listing the bits in the first column, followed by the bits in the second and so on, ending with  $A_{n-1,n}$ ) we get the following sequence

$$A_{12}, A_{13}, A_{23}, \dots, A_{1,n}, \dots, A_{n-1,n},$$

which can be seen to be nonstationary. Consider the case when  $n = 4, L = 2, Q_{11} = Q_{12} = 1, Q_{12} = 0$ . In this case the vertical ordering is

$$A_{12}, A_{13}, A_{23}, A_{14}, A_{24}, A_{34}$$

and this is seen to be nonstationary by observing  $P(A_{12} = 1, A_{13} = 0, A_{23} = 1) = 0$  but  $P(A_{14} = 1, A_{24} = 0, A_{34} = 1) > 0$ .

3. **Diagonally:** Consider  $\lfloor \frac{n}{2} \rfloor$  sequences defined as

$$\begin{aligned} S_1 &:= A_{12}, A_{23}, A_{34}, \dots, A_{n-1,n}, A_{n,1} \\ S_2 &:= A_{13}, A_{24}, A_{35}, \dots, A_{n-2,n}, A_{n-1,1}, A_{n,2} \\ &\vdots \\ S_{\lfloor \frac{n}{2} \rfloor - 1} &:= A_{1,1+\lfloor \frac{n}{2} \rfloor - 1}, A_{2,2+\lfloor \frac{n}{2} \rfloor - 1}, \dots, A_{n, \lfloor \frac{n}{2} \rfloor - 1} \end{aligned}$$

and

$$S_{\lfloor \frac{n}{2} \rfloor} = \begin{cases} A_{1,1+n/2}, A_{2,2+n/2}, \dots, A_{n/2,n}, & \text{when } n \text{ is even,} \\ A_{1,1+\lfloor \frac{n}{2} \rfloor}, A_{2,2+\lfloor \frac{n}{2} \rfloor}, \dots, A_{n,n+\lfloor \frac{n}{2} \rfloor}, & \text{when } n \text{ is odd.} \end{cases}$$

Concatenating  $S_1, \dots, S_{\lfloor \frac{n}{2} \rfloor}$  yields a sequence of length  $\binom{n}{2}$ . This corresponds to listing the bits diagonal-wise. However, even this does not yield a sequence that is stationary which can

be illustrated by considering the case when  $n = 4, L = 2, Q_{11} = Q_{12} = 1, Q_{13} = 0$ . In this case the diagonal ordering is

$$A_{12}, A_{23}, A_{34}, A_{41}, A_{13}, A_{24}$$

and this is seen to be nonstationary by observing  $P(A_{12} = 0, A_{23} = 1, A_{34} = 1) > 0$  but  $P(A_{34} = 0, A_{41} = 1, A_{13} = 1) = 0$ .

## 7 Experiments

We implement the proposed universal graph compressor (UGC) in four widely used benchmark graph datasets: protein-to-protein interaction network (PPI) [37], LiveJournal friendship network (Blogcatalog) [38], Flickr user network (Flickr) [38], and YouTube user network (YouTube) [39]. The block decomposition size  $k$  is chosen to be 1, 2, 3, 4 and we present in Table 1 the compression ratios (the ratio between output length and input length of the encoder) of UGC for different choices of  $k$ . We present in Table 2 the compression ratios of four competing algorithms.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
PPI	0.0228	<b>0.0226</b>	0.0227	0.034
Blogcatalog	0.0275	0.0270	<b>0.0267</b>	0.0288
Flickr	0.00960	0.00935	0.00915	<b>0.00907</b>
YouTube	$4.51 \times 10^{-5}$	$4.11 \times 10^{-5}$	<b><math>3.98 \times 10^{-5}</math></b>	$4.00 \times 10^{-5}$

Table 1: Compression ratio of UGC under different  $k$  values.

	CSR	Ligra+	LZ	PNG
PPI	0.166	0.0605	0.06	0.089
Blogcatalog	0.203	0.0682	0.080	0.096
Flickr	0.0584	0.0217	0.0307	0.0262
YouTube	$3.23 \times 10^{-4}$	$9.90 \times 10^{-5}$	$1.09 \times 10^{-4}$	$1.10 \times 10^{-3}$

Table 2: Compression ratios of competing algorithms.

- CSR: Compressed sparse row is a widely used sparse matrix representation format. In the experiment, we further optimize its default compressor exploiting the fact that the graph is simple and its adjacency matrix is symmetric with binary entries.
- Ligra+: This is another powerful sparse matrix representation format [40,41], which improves upon CSR using byte codes with run-length coding.
- LZ: This is an implementation of the algorithm proposed in [42], which first transforms the two-dimensional adjacency matrix into a one-dimensional sequence using the Peano–Hilbert space filling curve and then compresses the sequence using Lempel–Ziv 78 algorithm [23].
- PNG: The adjacency matrix of the graph is treated as a gray-scaled image and the PNG lossless image compressor is applied.

The compression ratios of the five algorithms implemented on four datasets are given as follows. The proposed UGC outperforms all competing algorithms in all datasets. The compression ratios from competing algorithms are 2.4 to 27 times that of the universal graph compressor.

Note, however, that CSR and Lagra+ are designed to enable fast computation, such as adjacency query or vertex degree query, in addition to compressing the matrix. Our proposed compressor does not possess such functionality and is designed solely for compression purpose.

## Acknowledgment

L. Wang would like to thank Emmanuel Abbe and Tsachy Weissman for stimulating discussions in the initial phase of the work. She is grateful to Young-Han Kim and Ofer Shayevitz for their interest and encouragement in this result. C. Wang would like to thank Richard Peng for his suggestion in writing.

## References

- [1] R. Rossi and R. Zhou, “GraphZIP: a clique-based sparse graph compression method,” *Journal of Big Data*, vol. 5, no. 10, 2018.
- [2] Y. Lim, U. Kang, and C. Faloutsos, “Slashburn: Graph compression and mining beyond caveman communities,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 3077–3089, 2014.
- [3] P. Boldi and S. Vigna, “The webgraph framework i: Compression techniques,” in *Proceedings of the 13th International Conference on World Wide Web*, ser. WWW ’04. New York, NY, USA: Association for Computing Machinery, 2004, pp. 595–602.
- [4] T. C. Conway and A. J. Bromage, “Succinct data structures for assembling large genomes,” *Bioinformatics*, vol. 27, no. 4, pp. 479–486, 01 2011.
- [5] M. Hayashida and T. Akutsu, “Comparing biological networks via graph compression,” *BMC systems biology*, vol. 4 Suppl 2, no. Suppl 2, 2010.
- [6] F. Chierichetti, R. Kumar, S. Lattanzi, M. Mitzenmacher, A. Panconesi, and P. Raghavan, “On compressing social networks,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 219–228.
- [7] G. Navarro, “Compressing web graphs like texts,” Dept. of Computer Science, University of Chile, Tech. Rep., 2007.
- [8] K. Sadakane, “New text indexing functionalities of the compressed suffix arrays,” *Journal of Algorithms*, vol. 48, no. 2, pp. 294 – 313, 2003.
- [9] N. R. Brisaboa, S. Ladra, and G. Navarro, “K2-trees for compact web graph representation,” in *Proceedings of the 16th International Symposium on String Processing and Information Retrieval*, ser. SPIRE ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 18–30.
- [10] A. Farzan and J. I. Munro, “Succinct encoding of arbitrary graphs,” *Theoretical Computer Science*, vol. 513, pp. 38 – 52, 2013.

- [11] G. Turán, “On the succinct representation of graphs,” *Discrete Applied Mathematics*, vol. 8, no. 3, pp. 289 – 294, 1984.
- [12] M. Naor, “Succinct representation of general unlabeled graphs,” *Discrete Applied Mathematics*, vol. 28, no. 3, pp. 303 – 307, 1990.
- [13] Y. Choi and W. Szpankowski, “Compression of graphical structures: Fundamental limits, algorithms, and experiments,” *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 620–638, Feb 2012.
- [14] P. Delgosha and V. Anantharam, “Universal lossless compression of graphical data,” in *Proc. IEEE Internat. Symp. Inf. Theory*, June 2017.
- [15] —, “Universal lossless compression of graphical data,” 2019.
- [16] —, “A universal low complexity compression algorithm for sparse marked graphs,” in *Proc. IEEE Internat. Symp. Inf. Theory*, June 2020.
- [17] E. Abbe, “Graph compression: The effect of clusters,” in *Proc. 54th Ann. Allerton Conf. Commun. Control Comput.*, 2016, pp. 1–8.
- [18] A. Asadi, E. Abbe, and S. Verdú, “Compressing data on graphs with clusters,” in *Proc. IEEE Internat. Symp. Inf. Theory*, August 2017, pp. 1583–1587.
- [19] M. Besta and T. Hoefler, “Survey and taxonomy of lossless graph compression and space-efficient graph representations,” 2018.
- [20] Q. Xie and A. R. Barron, “Minimax redundancy for the class of memoryless sources,” *IEEE Trans. Inf. Theory*, vol. 43, no. 2, pp. 646–657, 1997.
- [21] —, “Asymptotic minimax regret for data compression, gambling, and prediction,” *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 431–445, 2000.
- [22] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 337–343, 1977.
- [23] —, “Compression of individual sequences via variable-rate coding,” *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530–536, 1978.
- [24] M. Effros, K. Visweswariah, S. R. Kulkarni, and S. Verdú, “Universal lossless source coding with the burrows wheeler transform,” *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1061–1081, 2002.
- [25] F. M. Willems, Y. M. Shtarkov, and T. J. Tjalkens, “The context-tree weighting method: basic properties,” *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, 1995.
- [26] P. Delgosha and V. Anantharam, “Universal lossless compression of graphical data,” *IEEE Trans. Inf. Theory*, vol. 66, no. 11, pp. 6962–6976, 2020.
- [27] C. Bordenave and P. Caputo, “Large deviations of empirical neighborhood distribution in sparse random graphs,” *Probability Theory and Related Fields*, vol. 163, no. 1-2, p. 149–222, Nov 2014.
- [28] A. Bhatt, Z. Wang, C. Wang, and L. Wang, “Universal graph compression: Stochastic block models,” 2020. [Online]. Available: <https://arxiv.org/abs/2006.02643>

- [29] Y. Polyanskiy and Y. Wu, “Lecture notes on information theory,” 2014.
- [30] A. Frieze and M. Karoński, *Introduction to Random Graphs*. Cambridge University Press, 2015.
- [31] D. Marpe, H. Schwarz, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, 2003.
- [32] E. Abbe, A. S. Bandeira, and G. Hall, “Exact recovery in the stochastic block model,” *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 471–487, 2015.
- [33] E. Abbe and C. Sandon, “Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery,” in *IEEE 56th Annual Symposium on Foundations of Computer Science*, 2015, pp. 670–688.
- [34] E. Mossel, J. Neeman, and A. Sly, “Reconstruction and estimation in the planted partition model,” *Probability Theory and Related Fields*, vol. 162, no. 3-4, pp. 431–461, 2015.
- [35] S. Lauritzen, A. Rinaldo, and K. Sadeghi, “Random networks, graphical models, and exchangeability,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 80, 01 2017.
- [36] P. Billingsley, *Convergence of probability measures*, 2nd ed., ser. Wiley Series in Probability and Statistics: Probability and Statistics. New York: John Wiley & Sons Inc., 1999, a Wiley-Interscience Publication.
- [37] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 855–864.
- [38] L. Tang and H. Liu, “Relational learning via latent social dimensions,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 817–826.
- [39] S. Nandanwar and M. N. Murty, “Structural neighborhood based classification of nodes in a network,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1085–1094.
- [40] J. Shun and G. E. Blelloch, “Ligra: A lightweight graph processing framework for shared memory,” in *Proceedings of the 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ser. PPOPP ’13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 135–146.
- [41] J. Shun, L. Dhulipala, and G. E. Blelloch, “Smaller and faster: Parallel processing of compressed graphs with ligra+,” in *2015 Data Compression Conference*, pp. 403–412.
- [42] A. Lempel and J. Ziv, “Compression of two-dimensional data,” *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 2–8, 1986.