

# Representing Networks with 3D Shapes

Shengmin Jin  
Data Lab, EECS Department  
Syracuse University  
shengmin@data.syr.edu

Reza Zafarani  
Data Lab, EECS Department  
Syracuse University  
reza@data.syr.edu

**Abstract**—There has been a surge of interest in machine learning in graphs, as graphs and networks are ubiquitous across the globe and within science and engineering: road networks, power grids, protein-protein interaction networks, scientific collaboration networks, social networks, to name a few. Recent machine learning research has focused on efficient and effective ways to represent graph structure. Existing graph representation methods such as network embedding techniques learn to map a node (or a graph) to a vector in a low-dimensional vector space. However, the mapped values are often difficult to interpret, lacking information on the structure of the network or its subgraphs. Instead of using a low-dimensional vector to represent a graph, we propose to represent a network with a 3-dimensional shape: the *network shape*. We introduce the first network shape, a *Kronecker hull*, which represents a network as a 3D convex polyhedron using stochastic Kronecker graphs. We present a linear time algorithm to build Kronecker hulls. Network shapes provide a compact representation of networks that is easy to visualize and interpret. They captures various properties of not only the network, but also its subgraphs. For instance, they can provide the distribution of subgraphs within a network, e.g., what proportion of subgraphs are structurally similar to the whole network? Using experiments on real-world networks, we show how network shapes can be used in various applications, from computing similarity between two graphs (using the overlap between network shapes of two networks) to graph compression, where a graph with millions of nodes can be represented with a convex hull with less than 40 boundary points.

**Index Terms**—Network Shapes, Graph Representation, Kronecker Hulls, Network Convex Hull

## I. INTRODUCTION

Networks have become a universal language for describing complex data from science, engineering, and our daily life. Networks are used to study the role of a protein in biology [1], friendships in a social network [2], human emotions [3], among many other phenomena [4]. A compact, interpretable, visualizable, and efficient representation of networks facilitates scientific discoveries in a wide range of disciplines. Machine learning research aims to develop such network representations. Recent advancements in network representation, e.g., in network embedding [5]–[7] or latent representation learning [8], aim to learn a mapping from a (sub)graph, or its nodes, to points in a low-dimensional vector space. For example, a three node graph such as  $\triangle$  can be represented as a 2-dimensional vector: (1.24, 8.91). These techniques have shown remarkable performance in many applications, but face two fundamental limitations:

**I. Interpretability.** It is often difficult to understand the intuition behind learned representations. For instance, node (or subgraph) embedding techniques map nodes (or subgraphs) to points in a  $d$ -dimensional space, where no interpretation is often provided for such  $d$  dimensions. More specifically, when a graph is mapped to a point (a  $d$ -dimensional vector), one can hardly determine its exact structural properties from this vector, e.g., is it a dense network? The vector is mostly treated as a set of numeric features, limiting its usage.

**II. Preserving Subgraph Information.** As existing graph embedding approaches [9]–[11] map a network into a  $d$ -dimensional vector, the information on the subgraphs of this network are mostly aggregated, or lost. Hence, given the embedding for the whole network, it is challenging to identify how embeddings for its subgraphs would look like. One might hypothesize that for a network with billions of nodes, samples (i.e., subgraphs) that are close in size to the original network should have similar embeddings; however, for a small subgraph such as a triad  $\triangle$ , which is a subgraph of many networks, the embedding should not be necessarily similar to that of the original network. Statistically speaking, graph embedding is taking a sample from a network (i.e., a subgraph) and computing a *statistic* (i.e., an embedding) for that one sample, ignoring the *sampling distribution*: the distribution of embedding values for all subgraphs. We denote the distribution of embedding values for all subgraphs of a network as the network's *embedding space*. With a graph representation that can provide (1) the network's embedding space, or (2) means to approximate the embedding of a subgraph, e.g., using the embeddings of the whole network and/or some of its other subgraphs, one can preserve subgraph information.

**The Present Work: Network Shapes.** To address these limitations, we propose to represent a network as a set of vectors, representing the network and its subgraphs. These vectors will represent the embedding space of the network. By ensuring that these vectors are in a 3-dimensional space, and by identifying a 3D shape that contains all such 3D vectors, the network (and its subgraphs) can be represented as a 3D shape. We denote this shape as the *network shape*. We present the steps required to build network shapes, and the first algorithm for constructing network shapes. The algorithm is highly efficient, i.e., linear in the number of nodes and edges. The algorithm maps graphs into a 3D shape using stochastic Kronecker graphs and represents network shapes

using a convex hull, i.e., a convex polyhedron. We denote this network shape as the *Kronecker Hull* of the network. Overall, our contributions are mainly the following:

- 1) We propose network shapes, a 3D representation for a network that (i) is easy to interpret; (ii) captures various properties of not only the network, but also its subgraphs; (iii) facilitates easy network visualization; and (iv) enables various applications and comparative studies;
- 2) We propose Kronecker Hull, a network shape that represents a network and its subgraphs via a [convex] polyhedron in the three dimensional space;
- 3) We demonstrate how properties of a Kronecker hull (e.g., its volume or location) are connected to the structure of the network it represents. We study Kronecker hull properties using extensive experiments on eighteen real-world networks from four different categories; and
- 4) We show applications of network shapes in characterizing graphs (e.g., how does a 10% subgraph look like?), network categorization (e.g., is this a social or a biological network?), and computing graph similarity.

**Implications of Network Shapes.** Representing networks as 3D shapes has multiple benefits and applications:

- *Compact Representation of Networks.* Network shapes can help represent networks (and their embedding space) compactly. In most of our experiments, we can represent networks with million of nodes using shapes that can be represented with less than 40 boundary points.
- *Visualizing Networks.* Visualizing large graphs is challenging. This difficulty lies in the natural clutter, crossing, and overdrawing issues [12]. Network shapes help visualize networks (and their embedding space) with limited clutter.
- *Interpretation.* By properly designing network shapes, they can help illustrate structural properties of graphs and how a network is composed of subgraphs with different properties.
- *Features.* Features from network shapes such as their boundary points, center of gravity, volume, and other geometrical properties can capture various information about the network and its subgraphs and can be used for machine learning.
- *Applications.* Network shapes can be utilized in different applications, e.g., in computing graph similarity: The overlap of the shapes can indicate some level of similarity.

The rest of the paper is organized as follows. In Section II, we detail the necessary steps to build network shapes. In Section III, we discuss stochastic Kronecker graphs, the foundation behind Kronecker hulls (a network shape). Section IV provides the algorithm for computing the Kronecker Hull and its time complexity analysis. We summarize our experimental setup and data in Section V. With various experiments, we look into the properties of Kronecker hulls in Section VI. Section VII provides some applications which utilize network shapes. After reviewing additional related work in Section VIII, we conclude the paper in Section IX.

## II. BUILDING NETWORK SHAPES

The following simple steps can help build a network shape:

- Step 1: **Sample** many subgraphs from the network  
 Step 2: **Map** the network and its subgraphs to 3D vectors  
 Step 3: **Fit** a 3D Shape to the set of 3D vectors

The **first** requirement for constructing network shapes is a *sampling method*. Any sampling method can work. In our algorithm, we have utilized *Random Node Sampling* strategy [13]. Random node sampling uniformly at random selects a proportion  $p$  of nodes from a graph and the sample subgraph is then the graph induced by these selected nodes. Random node sampling is shown to perform well for various network measurements [13] and is a fast algorithm with linear time complexity. To sample systematically, one can sample by varying proportions of nodes (e.g., from 0% to 100%) with some fixed step size  $s$ . To control for variations, for each proportion, one can sample  $t$  independently sampled subgraphs, i.e., a total of  $t \times s$  subgraphs for one network.

The **second** requirement for constructing network shapes is an *embedding technique that can map a network to a 3D point*. The technique should provide embedding vectors that are easy to interpret and can capture the properties of the network and its subgraphs. Given such a technique, we can represent a network and its subgraphs as a set of 3D points. Similarly, one can think of many fast techniques to map a graph into a 3D vector, e.g., represent it with its (*diameter, average path length, clustering coefficient*). Here, we consider Stochastic Kronecker Graphs [14] as an appropriate candidate for mapping a graph into an interpretable 3D point, which we denote as the *Kronecker point*. In Section III, we investigate the properties and interpretation of Kronecker points.

The **third** and final requirement for building network shapes is a *technique to fit a 3D shape to a set of 3D points* obtained in Step 2 (3D embedding). While this can be done by fitting a variety of shapes (e.g., spheres), we consider building a network shape from a set of 3D points by computing its convex hull. A convex hull, for a set of points in a Euclidean space, is the smallest convex set that contains all the points in the original set [15]. Convex hull of a finite set of  $n$  points in a three-dimensional space can be computed with at most  $\mathcal{O}(n \log n)$  operations [16].

## III. STOCHASTIC KRONECKER GRAPHS

Stochastic Kronecker graphs [14] provide an approach to model large-scale graphs using the *Kronecker product*  $\otimes$  matrix operation. The Kronecker product generalizes matrix outer product, e.g., the Kronecker product of  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  and  $\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix}$ , denoted as  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix}$  is

$$\begin{aligned} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} &= \begin{bmatrix} 1 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 5 & 2 \cdot 0 & 2 \cdot 5 \\ 1 \cdot 6 & 1 \cdot 7 & 2 \cdot 6 & 2 \cdot 7 \\ 3 \cdot 0 & 3 \cdot 5 & 4 \cdot 0 & 4 \cdot 5 \\ 3 \cdot 6 & 3 \cdot 7 & 4 \cdot 6 & 4 \cdot 7 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}. \end{aligned}$$

When modeling a network using Stochastic Kronecker graphs, we aim to learn a small probability matrix  $P \in \mathbb{R}^{n \times n}$ , known as the *Kronecker initiator matrix*, such that the  $k^{\text{th}}$  Kronecker power of  $P$  (i.e.,  $P^{\otimes k} = \underbrace{P \otimes P \cdots \otimes P}_{k \text{ times}}$ ) is most

likely to have generated the adjacency matrix  $A \in \mathbb{R}^{n^k \times n^k}$  of the graphs which we are modeling, i.e.,  $P(A|P)$  is maximized (for further details refer to Ref. [14]). The KRONFIT algorithm can estimate the Kronecker initiator matrix for a real-world graph using maximum likelihood and in linear time [14].

#### A. Kronecker Points

Consider fitting a  $2 \times 2$  Kronecker initiator matrix  $I = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$  to a network. In an undirected network, where the adjacency matrix is symmetric, the Kronecker initiator matrix learned is also symmetric, i.e.,  $b = c$ . Thus, one can embed an undirected network, or any of its subgraphs, to a point  $(a, b, d)$  in the 3-D space, which we denote as the *Kronecker point* of a graph.

Kronecker points  $(a, b, d)$  have basic properties:

**I.** By definition, Kronecker initiator matrices are probability matrices, i.e., values  $a, b$  and  $d$  are all between 0 and 1. Hence, all possible graphs can be embedded in a  $1 \times 1 \times 1$  cube.

**II.** Without loss of generality, we can assume that  $a \geq d$ . Consider two initiator matrices  $\begin{bmatrix} a & b \\ b & d \end{bmatrix}$  and  $\begin{bmatrix} d & b \\ b & a \end{bmatrix}$  that only differ with respect to the positions of  $a$  and  $d$ , i.e., we can obtain one by simultaneous shuffling of rows and columns of the other based on some permutation. Calculating the  $k^{\text{th}}$  Kronecker power of both initiator matrices yields two adjacency matrices for two graphs. We can simply prove that these two graphs are indeed the same graph, i.e., the graphs are *isomorphic*. Assume  $P$  is a permutation matrix: a square binary matrix with exactly one entry of 1 in each row and column, and 0s elsewhere. Let  $X$  denote any initiator matrix. Then  $PXP^T$  represents a simultaneous shuffling of rows and columns of  $X$  according to permutation  $P$ . By Kronecker product properties  $(PXP^T)^{\otimes k} = P^{\otimes k} X^{\otimes k} (P^{\otimes k})^T$ . As  $P^{\otimes k}$  is also a permutation matrix, the graph represented by adjacency  $(PXP^T)^{\otimes k}$  is the same as the one by  $X^{\otimes k}$ .

#### B. Connections to network structure

One can interpret the  $2 \times 2$  initiator  $\begin{bmatrix} a & b \\ b & d \end{bmatrix}$  of an undirected network as a recursive expansion of two groups of network nodes into subgroups [14]. We can interpret  $a$  and  $d$  as the proportion of edges within each of the groups, and  $b$  as the proportion of edges between the two groups. As proved, we can assume that  $a \geq d$ ; hence, we can split the whole space into three regions, i.e., split all possible networks into three types. Each region represents a different network structure. We denote these regions as *Core-Periphery* ( $a \geq b \geq d$ ), *Dual-Core* ( $a \geq d \geq b$ ), and *Random* ( $b \geq a \geq d$ ).



#### Core-Periphery ( $a \geq b \geq d$ )

In networks with this configuration, at the high-level, the network can be divided into two groups. One group is dense with many connections as value  $a$  is the largest; fewer connections exist within the nodes in the other group as highlighted

by value  $d$ ; and moderate connections exist between nodes from different groups. Many real-world networks exhibit a core-periphery structure [17], where they form a core group and another group which acts as its periphery. Value  $a$  represents the *core strength*.



#### Dual-Core ( $a \geq d \geq b$ )

In this configuration, each group is internally well-connected but the connections between the two groups are sparse. We denote this configuration as the *Dual-Core* structure. Basically, the two groups of nodes form two major cores of the network, of which one exhibits a stronger core strength, and they are relatively independent of each other. Values  $a$  and  $d$  represent the core strength of each group.



#### Random ( $b \geq a \geq d$ )

This configuration is quite different from the previous two. Essentially, one can not find two recursive groups with more connections within each group than across groups. To some extent, it is indication that there is not much difference in the importance, or “coreness” among nodes. This reminds us of random graphs, such as those generated by the Erdős-Rényi  $G(n, p)$  model [18], where a random network of  $n$  nodes is created in which every edge exists with an equal probability  $p$ . To validate our speculation, we generate many random networks by fixing the number of nodes  $n = 1024$  and varying the probability  $p$ . We compute the Kronecker points of these networks to obtain  $a, b$ , and  $d$  values. Figure 1 illustrates that for a random network, we almost always have  $b \geq a \geq d$ , unless the graph is really dense, e.g,  $p > 0.75$ . We observe the same pattern for  $n = 2048, 4096$  and  $8192$ . One may note that when  $p$  is close to 0, the random network is empty, but value  $b$  does not converge to 0. This is an artifact caused by an overestimation in the KRONFIT algorithm, which we will detail when we discuss the limitations of KRONFIT.

For any graph or its subgraphs, their Kronecker points should be located within one of these three regions. This observation inspires us to represent a network using the location of the Kronecker points of the network and its subgraphs, e.g., a network that exhibits a core-periphery structure at the whole-network level, but most of its subgraphs are random graphs.

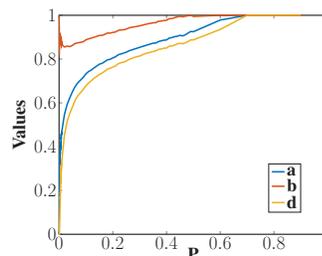


Fig. 1: Distribution of Kronecker points  $(a, b, d)$  for Random Networks  $G(n, p)$ . Here,  $n = 1024$ . We observe that in random graphs, unless the graph is really dense,  $b \geq a \geq d$ .

<p><b>Algorithm 1: KRONECKER HULL algorithm</b></p> <p><b>input</b> : an undirected network graph: <math>G(V, E)</math></p> <p><b>output</b> : the Kronecker hull of <math>G</math>: <math>KH_G</math></p> <p><b>parameter:</b> <math>s</math> : sampling proportion step size;  <math>t</math> : number of samples for one proportion;</p> <p>Kronecker_points = { };</p> <p><b>for</b> (<math>p = s</math>; <math>p &lt; 100\%</math>; <math>p = p + s</math>) {</p> <p>  <b>for</b> (<math>i = 1</math>; <math>i \leq t</math>; <math>i = i + 1</math>) {</p> <p>    %Sample a subgraph <math>G_p</math></p> <p>    <math>G_p = \text{RandomNodeSampling}(G, p)</math>;</p> <p>    %Fit Kronecker Initiator to <math>G_p</math></p> <p>    <math>\begin{bmatrix} a &amp; b \\ c &amp; d \end{bmatrix} = \text{KRONFIT}(G_s, 2)</math>;</p> <p>    Kronecker_point = (<math>a, b, d</math>);</p> <p>    Kronecker_points.add(Kronecker_point);</p> <p>  }</p> <p>}  <math>\begin{bmatrix} a &amp; b \\ c &amp; d \end{bmatrix} = \text{KRONFIT}(G, 2)</math>;</p> <p>Kronecker_point = (<math>a, b, d</math>);</p> <p>Kronecker_points.add(Kronecker_point);</p> <p><math>KH_G = \text{Quickhull}(\text{Kronecker\_points})</math>; %Convex Hull</p> <p><b>return</b> <math>KH_G</math>;</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### C. KRONFIT Limitations

KRONFIT can provide interpretable Kronecker points, but has a few limitations that may lead to over/underestimation. When the number of nodes within a real-world network is not a power of 2, KRONFIT will add isolated nodes so that the number of nodes becomes a power of 2 [19]. Adding isolated nodes may lead to underestimation of the parameters as it decreases the overall edge density and core strength of the groups. On the other hand, as the input to KRONFIT is a list of edges, when the network is extremely sparse and the graph size is small, KRONFIT can overestimate as it overlooks real isolated nodes within the network. The aforementioned overestimation in sparse random network fits this second case.

## IV. KRONECKER HULL

We introduce an algorithm to obtain the Kronecker hull of a network, and analyze its time complexity. The algorithm pseudocode is provided in Algorithm 1. The algorithm utilizes *Random Node Sampling* to sample many subgraphs from the network by (1) varying the proportion of nodes from 0% to 100% with step size  $s$  and (2) taking  $t$  independent samples for each proportion. For each sample (and the whole network), the algorithm computes its Kronecker point via KRONFIT algorithm. Finally, the convex hull of these Kronecker points are computed, using Quickhull algorithm [20], to obtain the Kronecker hull of the graph. The implementation is available at: <https://github.com/shengminjin/KroneckerHull>

**Time Complexity.** For one subgraph, random node sampling takes  $\mathcal{O}(n+m)$  and KRONFIT takes  $\mathcal{O}(n+m)$ , where  $|V| = n$  and  $|E| = m$ . Hence, for each subgraph, the time complexity is  $\mathcal{O}(n+m)$ . We have a total of  $\frac{100}{s} \times t + 1$  graphs (a network and its subgraphs) for which we compute Kronecker points. As the

number of Kronecker points is very small compared to the size of the network, the time spent on computing the convex hull is constant. Hence, the time complexity to compute Kronecker hull is  $\mathcal{O}(\frac{t}{s}(n+m))$ , linear in the number of nodes and edges.

## V. EXPERIMENTAL SETUP

For our experiments, we generate Kronecker hulls for various real-world networks by varying the proportion of nodes from 0% to 100% with step size 10%, i.e.,  $s = 10\%$  in Algorithm 1; for each proportion (except for 100% which represents the whole graph), we generate 20 independently sampled subgraphs, i.e.,  $t = 20$  in Algorithm 1. In total, we generate  $20 \times 9 + 1 = 181$  Kronecker points for each network, using which we obtain the Kronecker hull for the network. Next, we summarize the network data used in our experiments.

### A. Datasets

For our experiments, we use eighteen real-world networks from four general network categories: social networks, collaboration networks, road networks, and biological networks.

**Social Networks:** In total, we have eight social networks, and they are from three sub-categories.

#### ► Location-based Social Networks:

*Brightkite* and *Gowalla* [21]: were both once location-based social networking sites where users shared their locations by checking-in. Both networks were originally directed but have been converted to undirected where an undirected edge between users exist when friendships in both directions exist.

#### ► Friendship-based Social Networks:

*Hyves* [22]: the most popular social networking site in the Netherlands with mainly Dutch visitors. It competes with sites such as Facebook and MySpace in that country.

*Orkut* [21]: was a social networking website owned and operated by Google, shutdown in 2014.

*Livejournal* [23]: a social network where users can keep a blog or journal. Users can form friendship or follow others. In this dataset, edges represent friendships (undirected).

*MySpace* [23]: a social network with emphasis on music.

#### ► Video-Sharing or Movie Sites:

*YouTube* [21]: a video-sharing site with a social network.

*Flixster* [22]: a social movie site allowing users to buy, rent, or watch movies, share ratings, and discover new movies.

**Collaboration Networks:** We include four collaboration networks from arXiv.org, which include scientific collaborations between authors with different scientific interests. In a collaboration networks, an undirected edge between nodes  $i$  and  $j$  exists, if authors  $i$  and  $j$  have co-authored at least one paper.

*Astro-Ph* [21]: Astro physics.

*Cond-Mat* [21]: Condense matter physics.

*Gr-Qc* [21]: General relativity and quantum cosmology.

*Hep-Th* [21]: High energy physics theory.

**Road Networks:** We include three road networks. In road networks, nodes are intersections/endpoints and undirected edges are the roads connecting these intersections/road endpoints.

TABLE I: Dataset Statistics

Type	Network	$ V  = n$	$ E  = m$	Average Degree	Clustering Coefficient
Social Networks	Brightkite	58,228	214,078	7.353	0.1723
	Flixster	2,523,386	7,918,801	6.276	0.0834
	Gowalla	196,591	950,327	9.668	0.2367
	Hyves	1,402,673	2,777,419	3.960	0.0448
	Livejournal	3,017,286	85,654,976	56.776	0.1196
	MySpace	854,498	5,635,296	13.190	0.0433
	Orkut	3,072,441	117,185,083	76.281	0.1666
Collaboration Networks	YouTube	1,134,890	2,987,624	5.265	0.0808
	Astro-Ph	18,772	198,050	21.100	0.6306
	Cond-Mat	23,133	93,439	8.078	0.6334
	Gr-Qc	5,242	14,484	5.526	0.5296
Road Networks	Hep-Th	9,877	25,973	5.259	0.4714
	Road-CA	1,965,206	2,766,607	2.816	0.0464
	Road-PA	1,088,092	1,541,898	2.834	0.0465
	Road-TX	1,379,917	1,921,660	2.785	0.0470
Biological Networks	Bio-Dmela	7,393	25,569	6.917	0.0119
	Bio-Grid-Yeast	5,870	313,890	104	0.0516
	Human-Brain	177,600	15,669,036	176	0.4580

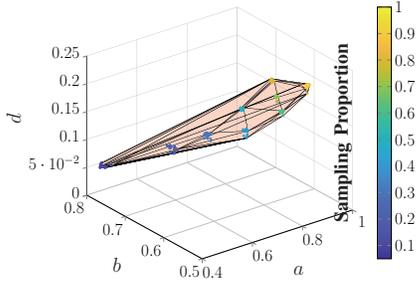


Fig. 2: Kronecker Hull for Hyves Social Network

*Road-CA* [21]: the road network of California.

*Road-PA* [21]: the road network of Pennsylvania.

*Road-TX* [21]: the road network of Texas.

**Biological Networks:** We include three biological networks.

*Bio-Grid-Yeast* and *Bio-Dmela* [21]: both protein-protein interaction networks.

*Human-Brain* [21]: the network of human brain.

The data statistics are summarized in Table I. To assess the impact of network structure on Kronecker hulls, for each real-world network, we generate a random synthetic network with a perturbed network structure, but with the same degree distribution, i.e., a *null model*. We create the null model using the configuration model [24], which can generate a random network with the same degree distribution and edge density (i.e.,  $|E|/\binom{|V|}{2}$ ) as the given real-world network.

## VI. KRONECKER HULL CHARACTERISTICS

To investigate the characteristics of Kronecker Hulls, we compute the Kronecker Hull for all networks. Figure 2 provides the Kronecker hull for one of our social networks: Hyves. The points on the boundary (or within) the Kronecker hull are Kronecker points  $(a, b, d)$  representing different sampling proportions. The Kronecker points are colored differently for different sampling proportions. We investigate different characteristics of Kronecker Hulls, but more importantly how the structure of a network is connected to those characteristics. In particular, we look at the volume, location, internal points, and boundaries of Kronecker hulls.

### A. Volume of Kronecker Hulls

As a Kronecker hull is a convex hull, its volume can be easily computed via triangulation. How is the volume of a Kronecker hull connected to the properties of the network it represents? Table II provides the volumes of the Kronecker hulls, denoted by  $\text{volume}(G)$ , for all networks. We observe that for social, road, and biological networks, volumes are between  $3.5 \times 10^{-5}$  to  $1.7 \times 10^{-3}$ . The maximum possible volume of a Kronecker hull can be 1 as values  $a$ ,  $b$ , and  $d$  lie in range  $[0, 1]$ . Hence, the Kronecker hulls of these networks are compact from a volume perspective, taking up only about one thousandth of the whole space. Volumes of collaboration networks are much larger, varying from  $3.4 \times 10^{-3}$  to 0.2, which we speculate is due to their specific network structure.

To investigate the impact of network structure on the volume of Kronecker hull, for any graph  $G$ , we compare the volume of its Kronecker hull  $\text{volume}(G)$  to that of its null model  $\text{volume}(G_{null})$ . Note that null models have the same edge density and degree distribution as the original graph, but with a random network structure. Hence, any change in volume indicates that network structure has an impact on volume. To compare volumes, we compute the ratio  $\frac{\text{volume}(G)}{\text{volume}(G_{null})}$ . While we observe that for all networks, the ratio is not equal to 1, indicating that network structure has an impact on the volume, the ratio often takes a value between 0.5 to 2, i.e., the actual volume can be at most twice, or at least half of that of its null model. We believe this finding can have implications in finding proper null models. As speculated, collaboration networks are outliers, with their network structures most damaged when constructing their null models: their Kronecker hull volume is much larger than that of their null models. Our further analysis indicated a strong correlation ( $\rho = 0.88$ ) between volume ratios and the clustering coefficient of networks, which is high in collaboration networks and is dramatically reduced in null models. We also conducted a multiple linear regression to predict volume based on five predictors:  $|V|$ ,  $|E|$ , edge density, average degree, and clustering coefficient. The regression coefficients also indicated that volume is strongly correlated to the edge density and clustering coefficient, with regression coefficients being nearly 0 for the other three variables.

### B. Location of Kronecker Hulls

To identify network properties that impact the location of a Kronecker hull, one must seek properties that when changed within a network, the new Kronecker hull for the modified network is at a different location in the 3D space, i.e., has less than 100% overlap with the original Kronecker hull. Hence, to investigate the impact of network structure on Kronecker hull location, we compute the overlap between Kronecker hulls of networks with that of their null models. We define the overlap between Kronecker hulls for networks  $A$  and  $B$  as

$$\text{overlap}(A, B) = \frac{\text{volume}(\text{KH}_A \cap \text{KH}_B)}{\min(\text{volume}(\text{KH}_A), \text{volume}(\text{KH}_B))}, \quad (1)$$

where  $\text{volume}$  is the volume of a Kronecker hull, and  $\text{KH}_A$  and  $\text{KH}_B$  represent Kronecker hulls of graphs  $A$  and  $B$ ,

TABLE II: Kronecker Hull Volume

Type	Networks	Edge Density ( $\times 10^{-4}$ )	Actual Graph		Null-Model		Ratio $\frac{\text{volume}(G)}{\text{volume}(G_{null})}$	overlap( $G, G_{null}$ )
			volume( $G$ ) ( $\times 10^{-4}$ )	Clustering Coefficient	volume( $G_{null}$ ) ( $\times 10^{-4}$ )	Clustering Coefficient		
Social Networks	Brightkite	1.262	6.22	0.1723	8.56	0.0053	0.73	70.49 %
	Flixster	0.025	3.42	0.0834	4.39	0.0012	0.78	83.87 %
	Gowalla	0.492	13.00	0.2367	7.86	0.0103	1.61	51.63 %
	Hypes	0.028	4.22	0.0448	5.52	0.0030	0.76	78.44 %
	Livejournal	0.188	1.93	0.1196	1.44	0.0013	1.34	75.69 %
	MySpace	0.154	3.75	0.0433	2.90	0.0037	1.29	86.37 %
	Orkut	0.248	0.44	0.1666	0.99	0.0006	0.45	26.38 %
YouTube	0.046	5.87	0.0808	5.94	0.0065	0.99	90.86 %	
Collaboration Networks	Astro-Ph	11.241	34.00	0.6306	3.89	0.0094	8.67	0 %
	Cond-Mat	3.492	98.00	0.6334	15.00	0.0022	6.36	16.99 %
	Gr-Qc	10.544	200.00	0.5296	15.00	0.0053	13.50	8.23 %
	Hep-Th	5.325	90.00	0.4714	13.00	0.0018	7.19	29.07 %
Road Networks	Road-CA	0.014	10.00	0.0464	11.00	$3.4 \times 10^{-7}$	0.89	87.57 %
	Road-PA	0.026	9.99	0.0465	9.72	$3.5 \times 10^{-6}$	1.03	72.89 %
	Road-TX	0.020	7.05	0.0470	8.31	0.0000	0.85	74.98 %
Biological Networks	Bio-Dmela	9.360	17.00	0.0119	20.00	0.0067	0.85	95.35 %
	Bio-Grid-Yeast	173.950	9.72	0.0516	10.00	0.0694	0.97	91.72 %
	Human-Brain	9.937	0.35	0.4580	0.10	0.0169	3.50	33.59 %

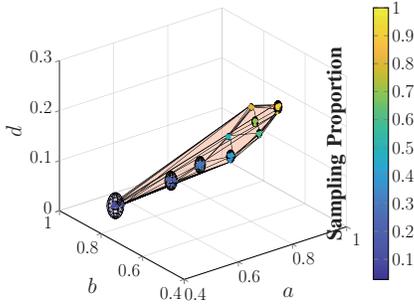


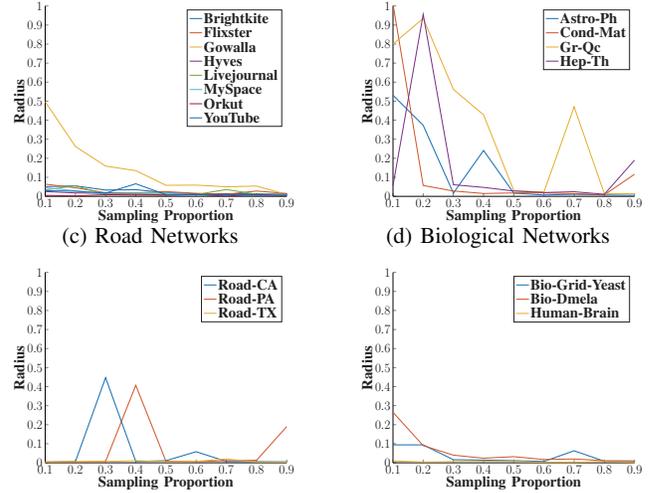
Fig. 3: Kronecker Hull of Hyves with Sphere fit

respectively. We define the overlap as a ratio: the volume of the intersection  $KH_A \cap KH_B$  normalized by the volume of the smaller Kronecker hull. It is easy to prove that given any collection of convex sets (finite, countable or uncountable), their intersection is a convex set. Therefore, the intersection of two Kronecker hulls  $KH_A \cap KH_B$  is also convex, allowing us to easily compute its volume. The results are in Table II. We observe an overlap that is less than 100% in all networks, indicating that network structure has an impact on the location of Kronecker hulls. Similar to our observations with respect to volume, (i) collaboration networks are outliers with very small overlaps and (ii) clustering coefficients of networks are strongly negatively correlated ( $\rho = -0.86$ ) to their overlaps. In addition, ratios  $\frac{\text{volume}(G)}{\text{volume}(G_{null})}$  are strongly negatively correlated ( $\rho = -0.77$ ) to overlaps  $\text{overlap}(G, G_{null})$  indicating that, e.g., when network structure is damaged, Kronecker hulls shrink in volume and move far from their original location.

### C. Internal Points

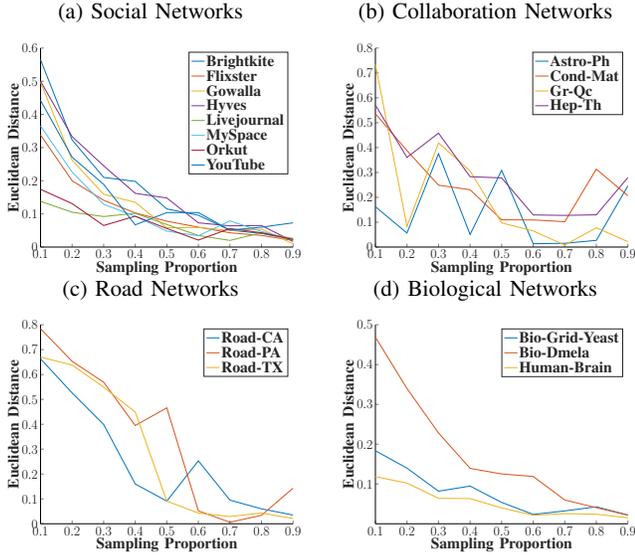
By definition, a point within a Kronecker hull of a network represents a sample from this network, i.e., a subgraph. Here, we investigate (1) how samples are distributed within a Kronecker hull and (2) how distances between samples are connected to similarities between corresponding subgraphs.

Fig. 4: Radius of Spheres Fit to Subgraph Kronecker Points  
(a) Social Networks (b) Collaboration Networks



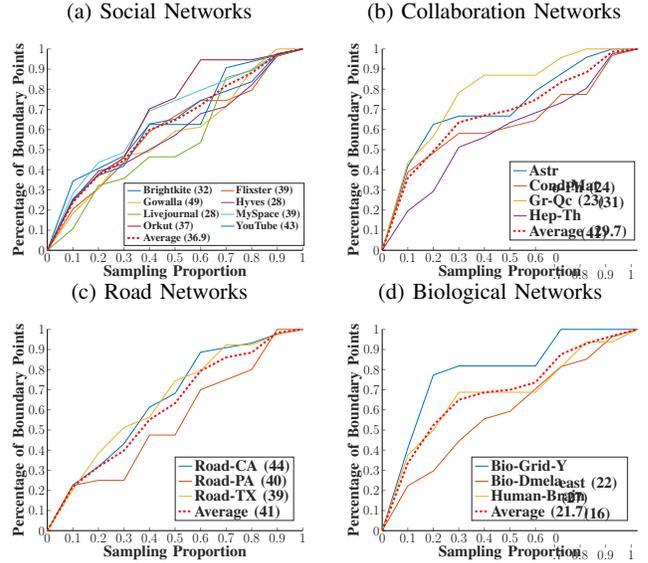
► *Sample Distribution.* In the Hyves example provided in Figure 2, a clustering phenomenon is observed: points representing samples of the same proportion appear to be clustered. To verify whether such a clustering exists, we fit a sphere to the points that represent the same sample size (see Figure 3). The sphere better visualizes the location of the cluster and its radius captures the variance. For all networks, we compute the radii of all such spheres; the results are in Figure 4. We find that the clustering phenomenon is observed for most networks, with relatively small radii that decreases as the sampling proportions increase. Compared to other networks, the radii of spheres of collaboration networks are larger, especially in smaller samples, i.e., clustering is not obvious. We speculate that this observation is due to samples being taken from different academic communities within the graph. Overall, our observations indicate that given a point within a Kronecker hull, nearby points are likely to be samples of the size.

Fig. 5: Distances between Sphere Centers (representing subgraph Kronecker points) and the Whole-graph Kronecker Point



► *Between-Sample Distances.* Consider two subgraphs of a network, each represented as a Kronecker point within the Kronecker hull of the network. Ideally, we hope that the distance between these two Kronecker points is related to the similarity between these two subgraphs. However, measuring similarity between two graphs can be challenging and subjective. To circumvent the challenge of computing graph similarity, we compute the distances between Kronecker points of graphs for which we have an intuitive understanding of their similarity. Here, we compute the distances between Kronecker points of different subgraphs and that of the whole network. This decision is based on the intuition that by increasing the sampling proportion, subgraphs should become more similar to the whole network (a 100% subgraph). As samples of the same proportion are clustered, we compute the Euclidean distance between the Kronecker point of the whole network and the sphere centers (representing Kronecker points for different sampling proportions). Figure 5 illustrates that with the increase in sampling proportion, sphere centers become closer to the Kronecker point of the whole network, indicating a convergence in Kronecker points as graphs become more similar. Looking at networks from different categories, we observe that (1) for social and biological networks, the distances drop fast when the sampling proportion increases from 10% to 30%, which suggests network structure of a 30% subgraph can be close to that of the whole network, when sampled using random node sampling; (2) for road networks, the sphere centers are far when the sampling proportion is small. With the increase in sampling proportion, the distances drop sharply when samples are below 60% and become very small after they reach 70%; (3) for collaboration networks, we observe a general decreasing trend in distances, but unlike other networks, there is an oscillation.

Fig. 6: Kronecker Hull Boundary Points Distribution. The numbers in the legend specify the number of boundary points.



#### D. Boundaries

Figure 6 provides the number (in the legend) and the distribution of boundary points of Kronecker hulls. We find that number of boundary points ranges from 16 to 49, and for most networks is between 30-40, out of the total of 181 points. Points from very small samples, especially those for sampling proportion 10%, are more likely to be boundary points. Points from middle size samples are more likely to be within the hull. Overall, we observe a continuity in points being on the boundary with the increase in sampling proportion. These findings suggest that (1) a limited number of points (e.g., 40) is required to store a Kronecker hull; (2) we can sample fewer points for each proportion to construct a Kronecker hull; and (3) boundary points can be used as compact features for machine learning on graphs.

### VII. APPLICATIONS

We present some applications of network shapes. In particular, we use Kronecker hulls to (A) describe a network and its subgraphs, (B) identify the category a network belongs to (e.g., road), and (C) study the similarity between two networks.

#### A. Characterizing Networks and their Subgraphs

As detailed in Section III, a Kronecker point, representing any graph, is guaranteed to fall within one of three regions: Core-Periphery, Dual-Core, and Random, where each region represents a specific network structure. This property allows one to describe the whole network, its subgraph(s), or a 3D space within its Kronecker hull.

We demonstrate this application by analyzing our networks. For each network, Table III provides the regions in which the whole graph and its 180 subgraphs are located. We make observations at the (I) whole-network or (II) subgraph levels:

TABLE III: Subgraphs of Networks. Subgraphs can exhibit a core-periphery structure , be dual-core , or random . Here, symbol  $\rightarrow$  indicates that the network structure observed (e.g., core-periphery) is the same as that of the smaller sampling proportion (to the left). A parenthesis is used to list all network structures observed at a sampling proportion.

Types	Sampling Proportion		10%	20%	30%	40%	50%	60%	70%	80%	90%	Whole Graph		
	Network													
Social Networks	Brightkite				$\rightarrow$									
	Flixster			$\rightarrow$										
	Gowalla				$\rightarrow$									
	Hyves				(   )		$\rightarrow$							
	Livejournal			$\rightarrow$										
	MySpace		(   )		$\rightarrow$									
	Orkut			$\rightarrow$										
	YouTube				$\rightarrow$									
Collaboration Networks	Astro-Ph		(   )	(   )		(   )			$\rightarrow$					
	Cond-Mat		(   )			$\rightarrow$						(   )		
	Gr-Qc		(   )	(    )	(    )	(   )			(   )					
	Hep-Th			(   )		$\rightarrow$				$\rightarrow$			(   )	
Road Networks	Road-CA				(   )		$\rightarrow$							
	Road-PA			$\rightarrow$			(   )			$\rightarrow$			(    )	
	Road-TX			$\rightarrow$				$\rightarrow$						
Biological Networks	Bio-Dmela			$\rightarrow$				$\rightarrow$						
	Bio-Grid-Yeast			$\rightarrow$										
	Human-Brain			$\rightarrow$										

*I. Characterizing Networks.* We identify region that the Kronecker point of the whole graph is within. We find that (1) all social networks are in the Core-Periphery region, confirming past research indicating that social networks exhibit a core-periphery structure [17]; (2) three collaboration networks are in the Core-Periphery region, and the other is in the Dual-Core region; (3) all road networks are within the Dual-Core region, which can be explained by the fact that road networks often exhibit a recursive structure. For example, the connections between two states are sparse, relying on a few highways or trunk roads, while the connections within a state are denser. This road structure also applies to two cities within a state; (4) all biological networks are in the Core-Periphery region, confirming past research that has observed a core-periphery structure within protein-protein interaction networks [25] and human brain [26]; and (5) no network is in the Random region.

*II. Characterizing Subgraphs.* By identifying the regions for subgraphs, we find that: (1) for social and biological networks, most subgraphs are in the same region in which the whole

network is in: the Core-Periphery region. This observation indicates that small samples (e.g., 20%) of most social and biological networks exhibit properties similar to that of the whole network. This observation also explains our previous observation on the rapid drop of distances between sphere centers and the Kronecker point of the whole network when the sampling proportion changes from 10% to 30%. We also observe that when the sample is too small, the network core is not yet formed in some samples, leading to those samples being in the Random region; (2) for road networks, we find sampled subgraphs that are less than 50% of the network are often in the Random region, and after that exhibit a Dual-Core structure. This transition explains why the distances between sphere centers and the Kronecker point of the whole network drop sharply when the proportion is less than 60%; (3) for collaboration networks, the composition of subgraphs is complex. For large samples, subgraphs exhibit either a Core-Periphery or a Dual-Core structure. For small samples, we also observe some Random subgraphs. Also, subgraph structure

Fig. 7: Kronecker hulls across Categories

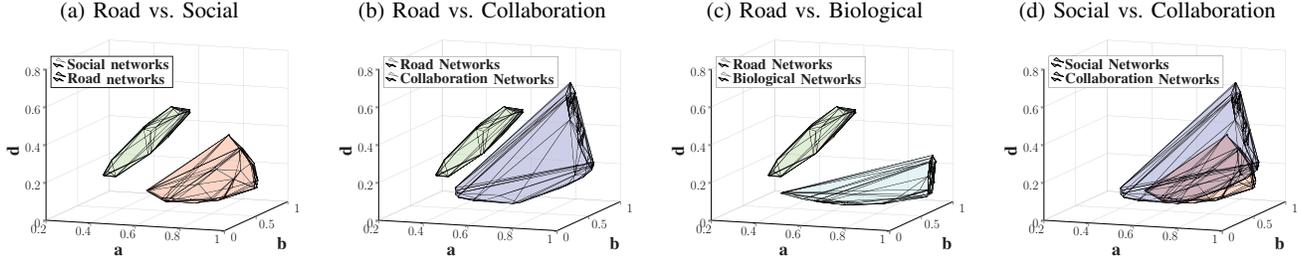


TABLE IV: Overlap between Kronecker Hulls of Categories

	Social Networks	Collaboration Networks	Biological Networks
Social Networks	100%	75.63%	8.92%
Collaboration Networks	75.63%	100%	4.22%
Biological Networks	8.92%	4.22%	100%

TABLE V: Kronecker Hull Overlaps for Social Networks

	Brightkite	Flixster	Gowalla	Hypes	Livejournal	MySpace	Orkut	YouTube
Brightkite	100%	0.07%	12.2%	0	0	0	0	0
Flixster	0.07%	100%	50.14%	0	0	9.25%	0	49.03%
Gowalla	12.2%	50.14%	100%	0	10.56%	0	0	11.64%
Hypes	0	0	0	100%	0	0	0	0
Livejournal	0	0	10.56%	0	100%	0	0	0
MySpace	0	9.25%	0	0	0	100%	0	16.52%
Orkut	0	0	0	0	0	0	100%	0
YouTube	0	49.03%	11.64%	0	0	16.52%	0	100%

strongly depends on sampled nodes. This complexity explains why Kronecker points do not cluster well in collaboration networks as same-size samples can exhibit various network structures, e.g., for being from various academic communities.

### B. Network Categorization

Kronecker hulls can help categorize networks, i.e., determine whether a network is a social network or a biological one. We demonstrate the feasibility of network categorization using Kronecker hulls. To categorize networks, we create a Kronecker hull for a family of graphs (e.g., all social networks). Here, for each network category (biological/social/road/collaboration), we create a Kronecker hull from the Kronecker points (i.e., subgraphs) of all the networks within that category. As depicted in Figure 7, Kronecker hull of road networks is well-separated from those of the other three categories. Basically, given a Kronecker hull of one road network, or Kronecker points of some subgraphs from a road network, one can easily verify that it is not from the other three categories. For the other three categories, we compute the overlap between their corresponding Kronecker hulls. From Table IV, we find that biological networks have a small overlap with the other two types of networks, meaning that it is not very difficult to distinguish a biological network from a social or a collaboration network. However, the overlap between collaboration networks and social networks is large, being over 75%. We plot both Kronecker hulls in Figure 7d. This large overlap is not surprising, as both categories involve human social behavior. Clearly, a comprehensive supervised learning framework (e.g., that uses Kronecker hull attributes as features) can further advance network categorization.

### C. Computing Network Similarity

Kronecker hulls can capture various forms of (dis) similarity between two networks:

**I.** Consider two large graphs  $A$  and  $B$  to be 100% similar when  $A$  is a subgraph of  $B$ . By construction, Kronecker hull of  $A$  will be within Kronecker hull of  $B$ , i.e., a 100% similarity leads to 100% overlap between the corresponding Kronecker hulls. Hence, the overlap may indicate some level of similarity.

**II.** Consider two graphs to be similar, when they both belong to similar categories of networks (e.g., a social network is similar to a collaboration network) and dissimilar, otherwise. Our discussion in Section VII-B showed that when networks belong to dissimilar categories, there is little to no overlap between their Kronecker hulls. For instance, a road network in our dataset will have no overlap with a random network from any other category, while a social network is expected to have some overlap with a collaboration network.

**III.** Consider two networks to be similar, when they are semantically similar, e.g., both are video sharing networks. Here, we assume semantic similarity leads to some level of network structure similarity. We show that Kronecker hulls can capture some level of semantic (dis) similarity by taking social networks as an example. Table V lists the overlap between the Kronecker hulls of each pair of the eight social networks. We make the following observations: (1) various similar networks exhibit overlap. For example, Brightkite and Gowalla, both location-based social networks, overlap. Also, MySpace, YouTube and Flixster are well connected to each other, which may be explained by the content they share. MySpace has a strong music emphasis, and YouTube and Flixster are often used to share videos or music; and (2) social networks popular in specific countries (e.g., Orkut and Hypes) are well separated from other networks.

We believe these observations motivate a systematic study on the connection between graph similarity and overlap of Kronecker hulls, which we leave as part of our future work.

### VIII. RELATED WORK

In addition to related research discussed throughout the paper, our work has links to the following areas:

**I. Network Visualization.** Network visualization [27] aims to visualize large-scale networks in real-time to facilitate easy network exploration or specific applications, e.g., detecting

users with expertise [28]. Network shapes provide a compact and interpretable way to visualize a network and its subgraphs.

**II. Graph Compression.** There has been an increasing interest in graph compression [29]–[31], especially in large-scale real-world networks. Storing the network shape provides an alternative compact solution to graph compression. In our experiments on graphs with millions of nodes, Kronecker hulls can often be represented with less than 40 boundary points.

## IX. CONCLUSIONS

We propose network shapes and a linear algorithm to construct one type of network shapes: Kronecker Hulls. A Kronecker hull represents a network as a convex hull. Kronecker hulls are compact, easy to visualize, and capture various properties of a network and its subgraphs. Kronecker hulls can be used in applications such as categorizing graph (e.g., is the network biological or social?) or to assess graph similarity.

Our study could be extended, empirically or theoretically, by designing other types of network shapes, i.e., by extending the three general steps of network shapes: (1) sampling a graph, (2) mapping a graph to a 3D point, and (3) fitting a shape to a set of 3D points. For sampling, we use random node sampling. Investigating random edge sampling or random walks may lead to shapes that capture different network properties. For mapping a graph to a 3D point, we use stochastic Kronecker graphs. One can investigate other embedding techniques such as [9], [32], [33], or design other shape-specific embedding techniques with theoretical guarantees. Finally, we represent shapes as convex hulls. Other compact means to represent shapes, e.g. spheres, may enable further applications.

As for applications, exploring the possibility of using network shapes in various network tasks, e.g., community detection or anomaly detection may lead to interesting discoveries.

## REFERENCES

- [1] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1025–1035.
- [2] L. Backstrom and J. Leskovec, “Supervised random walks: predicting and recommending links in social networks,” in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 635–644.
- [3] S. Jin and R. Zafarani, “Emotions in social networks: Distributions, patterns, and models,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1907–1916.
- [4] M. Newman, *Networks*. Oxford university press, 2018.
- [5] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD conference*. ACM, 2016, pp. 855–864.
- [6] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.
- [7] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 1067–1077.
- [8] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: Methods and applications,” *arXiv preprint arXiv:1709.05584*, 2017.
- [9] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” in *Advances in neural information processing systems*, 2015, pp. 2224–2232.
- [10] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.
- [11] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” *arXiv preprint arXiv:1312.6203*, 2013.
- [12] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner, “Visual analysis of large graphs: state-of-the-art and future research challenges,” in *Computer graphics forum*, vol. 30, no. 6. Wiley Online Library, 2011, pp. 1719–1749.
- [13] J. Leskovec and C. Faloutsos, “Sampling from large graphs,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 631–636.
- [14] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, “Kronecker graphs: An approach to modeling networks,” *JMLR*, vol. 11, no. Feb, pp. 985–1042, 2010.
- [15] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, “Computational geometry,” in *Computational geometry*. Springer, 2000, pp. 1–17.
- [16] F. P. Preparata and S. J. Hong, “Convex hulls of finite sets of points in two and three dimensions,” *Communications of the ACM*, vol. 20, no. 2, pp. 87–93, 1977.
- [17] S. P. Borgatti and M. G. Everett, “Models of core/periphery structures,” *Soc. networks*, vol. 21, no. 4, pp. 375–395, 2000.
- [18] P. Erdős and A. Rényi, “On random graphs, i,” *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.
- [19] J. Leskovec and C. Faloutsos, “Scalable modeling of real graphs using kronecker multiplication,” in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 497–504.
- [20] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The quickhull algorithm for convex hulls,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.
- [21] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” <http://snap.stanford.edu/data>, Jun. 2014.
- [22] R. Zafarani and H. Liu, “Social computing data repository at ASU,” 2009. [Online]. Available: <http://socialcomputing.asu.edu>
- [23] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, “Cosnet: Connecting heterogeneous social networks with local and global consistency,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1485–1494.
- [24] M. E. Newman, “The structure and function of complex networks,” *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.
- [25] F. Luo, B. Li, X.-F. Wan, and R. H. Scheuermann, “Core and periphery structures in protein interaction networks,” in *Bmc Bioinformatics*, vol. 10, no. 4. BioMed Central, 2009, p. S8.
- [26] D. S. Bassett, N. F. Wymbs, M. P. Rombach, M. A. Porter, P. J. Mucha, and S. T. Grafton, “Task-based core-periphery organization of human brain dynamics,” *PLoS computational biology*, vol. 9, no. 9, p. e1003171, 2013.
- [27] M. Bastian, S. Heymann, M. Jacomy *et al.*, “Gephi: an open source software for exploring and manipulating networks.” *Icswm*, vol. 8, pp. 361–362, 2009.
- [28] J. Zhang, M. S. Ackerman, and L. Adamic, “Expertise networks in online communities: Structure and algorithms,” in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW ’07. New York, NY, USA: ACM, 2007, pp. 221–230. [Online]. Available: <http://doi.acm.org/10.1145/1242572.1242603>
- [29] T. Feder and R. Motwani, “Clique partitions, graph compression and speeding-up algorithms,” in *Proceedings of the twenty-third annual ACM symposium on Theory of computing*. ACM, 1991, pp. 123–133.
- [30] A. Apostolico and G. Drovandi, “Graph compression by bfs,” *Algorithms*, vol. 2, no. 3, pp. 1031–1044, 2009.
- [31] C. A. Packer and L. B. Holder, “Graphzip: Dictionary-based compression for mining graph streams,” *arXiv preprint arXiv:1703.08614*, 2017.
- [32] P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey,” *arXiv preprint arXiv:1705.02801*, 2017.
- [33] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, “Graph kernels,” *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1201–1242, 2010.