The Rise and Fall of Network Stars: Analyzing 2.5 Million Graphs to Reveal How High-Degree Vertices Emerge over Time

Michael Fire[1,2]* and Carlos Guestrin[1]

Abstract
Trends change rapidly in today's world, prompting this key question: What is the mechanism behind the emergence of new trends? By representing real-world dynamic systems as complex networks, the emergence of new trends can be symbolized by vertices that "shine." That is, at a specific time interval in a network's life, certain vertices become increasingly connected to other vertices. This process creates new high-degree vertices, i.e., network stars. Thus, to study trends, we must look at how networks evolve over time and determine how the stars behave. In our research, we constructed the largest publicly available network evolution dataset to date, which contains 38,000 real-world networks and 2.5 million graphs. Then, we performed the first precise wide-scale analysis of the evolution of networks with various scales. Three primary observations resulted: (a) links are most prevalent among vertices that join a network at a similar time; (b) the rate that new vertices join a network is a central factor in molding a network's topology; and (c) the emergence of network stars (high-degree vertices) is correlated with fast-growing networks. We applied our learnings to develop a flexible network-generation model based on large-scale, real-world data. This model gives a better understanding of how stars rise and fall within networks, and is applicable to dynamic systems both in nature and society.

Multimedia Links
► Video ► Interactive Data Visualization ► Data ► Code Tutorials

Keywords
Data Science; Network Science; Big Data; Complex Network Evolution; Complex Networks Models; Networks Dataset

[1] *Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA*
[2] *The eScience Institute, University of Washington, Seattle, WA*
***Corresponding author**: fire@cs.washington.edu

Contents

1. Introduction

Change is inevitable, yet the mechanisms behind changing trends are not well understood [1, 2]. By investigating these mechanisms, we can better answer questions such as how people gain and lose political power, why some companies thrive while others shrivel, and how infectious diseases patterns can spread throughout populations. To study the mechanisms that influence new trends, we can represent various dynamic systems as complex networks and then explore how these networks change over time. Complex networks are loosely defined as networks with non-trivial structure and dynamics, appearing in many real-world systems [3, 4, 5]. Networks consist of a set of vertices and a set of links connecting these vertices. Vertices can represent a wide range of entities, such as online social network users [8], neurons [9], or proteins [1, 7]. The popularity of a vertex can be measured by the number of links connected to it from other vertices in the network, where the links can be directed, like in Twitter[1] where one user follows another user, or undirected, like a mutual friendship between two people [8]. The most popular vertices–vertices with many connections–are referred to as stars. The objective of our research was to use large-scale, real-world data to better understand how real-world networks evolve over long periods of time. We then narrowed that objective to study network stars and gain significant insights into their rise and fall over months, years, and even centuries as networks evolve.

We utilized a variety of large-scale datasets, data science tools, and extensive cloud computing resources to assemble the world's largest complex network evolution dataset. The dataset consists of billions of records used to construct and analyze the evolution process of over 38,000 complex networks and the topological properties of more than 2.5 million graphs over long periods of times (see Table 1 and Section 3.1). Namely, we constructed and analyzed the following networks:

- *Citation and co-authorship networks*, created from the Microsoft Academic Graph [10], which includes more than 126 million papers and 114 million authors over a period of 215 years.
- The Reddit social network, created from over 2.71 billion comments over a period of more than 10 years [11].
- *Chess players network*, created from over 214 million games during a period of 18 years [12].
- *People marriage network*, created from the WikiTree online genealogy dataset, including 1.96 million marriage records over 610 years [13].
- *Bitcoin network*, created by over 37 million Bitcoin transactions over a period of 4 years [14].

In addition to analyzing the large complex networks described above, we analyzed the evolution of about 18,000 co-authorship and citation networks of various research fields. We also analyzed the evolution of more than 20,000 communities for the Reddit dataset (see Section 3.1.1).

We utilized the constructed extensive dataset to perform the first precise wide-scale analysis of the evolution of networks with various scales. By examining the evolution and dynamics of these networks, three notable observations emerged: First, links are most prevalent among vertices that join a network at a similar time (see Figure 1). For example, in the citation network, over 80% of all citations referenced publications published within 15 years, while less than 8% had a publication gap of more than 25 years. Similarly, in the WikiTree network, 69.2% and 8.2% of married couples had age differences of fewer than 7 years or over 15 years, respectively.

Second, the rate that new vertices join a network is a central factor in molding a network's topology. We identified six common patterns in which vertices tend to join the networks (see Figure 2). Moreover, we observed that different vertex-join patterns influence the structures and properties of the networks (see Figures 4 and 6). For example, we identified that on average fast-growing networks tend be active longer, have more vertices, be less dense, and cluster less than slow-growing networks.

Third, network stars (high-degree vertices) tend to emerge in networks that are growing rapidly. For slow-growing networks, most stars emerged a short time after the network became active and kept their place, while for fast-growing networks, stars emerged at any time (see Figure 5).

We applied our learnings to develop a straightforward random network-generation model that more accurately depicts how networks evolve (see Figures 7 and 8, and Table 2). Our Temporal Preferential Attachment (TPA) model improves upon previous models because it more correctly represents real-world data, especially for networks that are growing quickly, and can be used in a more flexible manner. Furthermore, our model can give insights on the changing popularity of network stars.

This study has several contributions. To our knowledge, this is the largest study–by several orders of magnitude–to analyze real-world complex networks over long periods of times. The key contributions presented in this paper are fivefold:

1. We determined that different users' arrival patterns are a significant factor in the molding of a network's topology. The evidence we present shows that different vertex-joining patterns tend to create communities with different structures.
2. In all the analyzed networks from different domains, we observed that time is a crucial factor in the way a network evolves. Vertices tend to connect to other vertices that join the network at a similar time.
3. By carefully analyzing the topological properties of over 2.5 million graphs, we discovered certain rules

[1] http://twitter.com

Table 1. Network Datasets.

Network	Graph Type	Vertices Number	Edges Number	Time Period	Analyzed Networks
Citations	Directed	126,903,970	528,682,289	215 years	8,996 networks; 769,793 graphs
Coauthorship	Undirected	114,697,977	6,706,308,601	215 years	9,005 networks; 770,854 graphs
Reddit	Directed	20,298,899	991,531,578	568 weeks	20,128 networks; 1,023,995 graphs
Chess Games	Undirected	519,583	74,673,247	18 years	-
WikiTree Marriages	Undirected	3,723,557	1,959,540	610 years	-
Bitcoin Transactions	Directed	6,336,769	16,057,711	222 weeks	-

govern the way key entities in a network gain power. This can help explain a wide range of changing trends, from the results of the 2016 US elections to how the Kardashian family became mega celebrities on Instagram.[2]

4. We developed a simple model, utilizing all the above observations, that uses real-world big data to confirm and explain our observations. A model must accurately reflect real life.

5. We constructed the largest network evolution corpora that is publicly available. This dataset can immensely aid researchers in investigating and understanding complex dynamic systems.

The remainder of the paper is organized as follows: In Section 2, we provide an overview of various related studies. In Section 3, we describe the datasets, methods, algorithms, and experiments used throughout this study. Next, in Section 4, we present the results of our study. Afterwards, in Section 5, we present our TPA model. Then, in Section 6, we discuss the obtained results. Lastly, in Section 7, we present our conclusions from this study and also offer future research directions.

2. Related Work

The study of complex networks began over half a century ago, in 1965. While studying a network of citations among scientific papers, Price observed a network in which the degree distribution followed a power law [15]. Later, in 1976, Price [16] provided an explanation of the creation of these types of networks: "Success seems to breed success. A paper which has been cited many times is more likely to be cited again than one which has been little cited" [16]. Price subsequently offered a method for the creation of networks in which the degree distribution follows a power law.

Several decades later, Watts and Strogatz [17] and Newman and Watts [**?**] introduced models for generating small-world networks. Typically, small-world networks have a relatively high clustering coefficient, and the distance between any two vertices scales as the logarithm of the number of vertices [18]. Barabási and Albert observed that degree distributions that follow power laws exist in a variety of networks, including the World Wide Web [6]. Barabási and Albert coined the term "scale-free networks" for describing such networks. Similar to Price's method [16], Barabási and Albert [6] suggested a simple and elegant model for creating random complex networks based on the rule that the rich are getting richer. In the BA model, a network starts with m connected vertices. Each new vertex that is added (one at a time) has a greater probability of connecting to pre-existing vertices with higher degree, where the probability of connecting to an existing vertex is proportional to vertex's degree [6]. Consequently, rich vertices with high degrees tend to become even richer due to their connections with new vertices that join the graph. Many real-world complex networks have a community structure in which "the division of network nodes into groups within which the network connections are dense, but between which they are sparser" [19]. In 2000, Dorogovtsev et al. [20] suggested a model with preferential linking that takes into consideration a vertex attractiveness. In 2002, Holme and Kim [21] extended the Barabási and Albert model to include a "triad formation step." The Holme and Kim model creates networks with both the perfect power-law degree distribution and high clustering. In 2004, Newman and Girvan proposed a community detection algorithm and offered a simple method to create networks with community structure [19]. In 2007, Leskovec et al. [**?**] introduced the "forest fire" graph generation model, based on a "forest fire" spreading process.

[2] https://www.instagram.com/kimkardashian

Even though the models described above can explain some of the characteristics of real-world complex networks, the random networks created by these models were lacking in other properties that were observed in real-world complex networks. Therefore, in recent years, other models have been suggested which have additional characteristics [17, 18, 21]. Thorough reviews on complex networks and complex network evolution models can be found in books by Chung and Lu [**?**], Newman et al. [22], and by Dorogovtsev and Mendes [23].

A similar study to ours was conducted by Leskovec et al. [24]. They performed edge-by-edge analysis of four large-scale networks – Flickr, Delicious, LinkedIn, and Yahoo Answers – with time spans ranging from four months to almost four years. By studying a wide variety of network formation strategies, they observed that edge locality plays a critical role in the evolution of networks, and they offered a model which focused on microscopic vertex behavior. In their proposed model, vertices arrive at a pre-specified rate and choose their lifetimes. Afterwards, each vertex "independently initiates edges according to a 'gap' process, selecting a destination for each edge according to a simple triangle-closing model free of any parameters" [24]. They showed that their model could closely mimic the macroscopic characteristics of real social networks. Additionally, Leskovec et al., similar to our study, observed the arrival patterns of various vertices. Namely, they observed that (a) Flickr's network data has grown exponentially; (b) Delicious has grown slightly superlinearly; (c) LinkedIn has grown quadratically; and (d) Yahoo Answers has grown sublinearly. Due to these observations, they concluded that vertex arrival functions needed to be part of their proposed model. However, their study did not analyze the implications of using different arrival functions.

The body of literature has increased extensively over the last two decades, with hundreds of new network studies each year,[3] and many papers present observations and network models that overlap with this study. To the best of our knowledge, however, this study is the first to present a general model based on extensive analysis of large-scale real data utilizing over 38,000 real-world complex networks. Moreover, this study is the first to utilize extensive temporal complex network data to understand how high-degree vertices emerge over time.

3. Methods and Experiments

3.1 Constructing the Network Datasets

In this study, we utilized six different datasets to construct various types of networks. Below we describe in detail how we generated the complex network corpora with over 38,000 networks.

3.1.1 The Reddit Networks

Reddit is a news aggregation website and online social platform launched in 2005 by Steve Huffman and Alexis Ohanian [25]. Reddit users (also known as "redditors") can submit content on the website, which is then commented upon, and upvoted or downvoted by other users in order to increase or decrease the submission visibility. Redditors can also create their own subreddit on a topic of their choosing, make it public or private, and let other redditors join it. This makes Reddit a collection of online communities centered around a variety of topics such as books, gaming, science, and asking questions In this study, we utilized the Reddit dataset which was recently made public by Jason Michael Baumgartner [11]. Specifically, we utilized over 2.71 billion comments that were posted from December 2005 through October 2016. These posts were created by 20,299,812 users with unique usernames in 416,729 different subreddits. The dataset contains information on the exact time and date each comment was posted. Moreover, the dataset contains each comment's ID, as well as information on the user who posted it and the ID of the parent comment, i.e., the ID to which the current comment replied. We cleaned the dataset by removing nonessential comments, specifically those that were marked as deleted and those that did not include the information of the user who posted them. Additionally, we removed posts by users who with high probability were bots. Namely, we removed all the users who posted more than 100,000 comments each, and we removed redditors whose comments appeared in the bots list published in the BotWatchman subreddit.[4] We downloaded the bots list from the BotWatchman subreddit during November 2016. After the removal of these posts, we were left with over 2.39 billion comments published in 371,841 subreddits by 20,298,899 users.

Next, we constructed social networks from the subreddits' comments data. However, many of the subreddits did not contain enough comments. Therefore, for all the subreddits in the clean dataset with about 2.39 billion comments, we selected only those subreddits that had at least 1,000 comments and more than a single user. Out of all the subreddits, 20,145 fulfilled these criteria, out of which we succeeded in constructing the social networks over time of 20,136 subreddits with over 2.37 billion posts (referred to as selected subreddits). Afterwards, for each selected subreddit, similar to the construction method used by Kairam et al. [26], we created the subreddit's social network directed graph by connecting users who posted comments as replies to other posted comments.

Namely, for a subreddit S, we define the subreddit's directed graph at time t to be: $G_t^S := < V_t^S, E_t^S >$, where V_t^S is the set of vertices representing all the subreddit's users who posted at least a single comment in the subreddit up to t days after the subreddit became active, i.e., when the first comment was published in the subreddit S. In addition, $e := (u,v) \in E_t^S$ is the list of all edges between the subreddit's users, $u \in V_t^S$ and $v \in V_t^S$, created up to t days after the subreddit became active. We define an edge between u and v to exist if there exists a comment on the subreddit posted by u to which v posted a reply on the same subreddit. Lastly, to better understand

[3] According to Google Scholar over 870 papers' titles published in 2016 contains the phrase "complex networks."

[4] https://www.reddit.com/r/BotWatchman/

how subreddits evolve over time, for each selected subreddit S, we created a set of incremental graphs in incremental time intervals of every 4 weeks between the time the subreddit initially became active and the time the last comment was posted in the subreddit according to the dataset. Overall, we created over a million graphs that contain detailed information on how these selected subreddits evolved over time.

It is important to notice that the constructed directed graphs also include single vertices of redditors who posted comments and did not receive any reply, as well as self-loop edges of redditors who posted a comment and then posted a reply to their own comment.

3.1.2 The Free Internet Chess Server Network

The Free Internet Chess Server (FICS)[5] is one of the oldest and largest Internet chess servers. The FICS serves over 540,000 users who have played over 300 million chess games [12]. For this study, we downloaded the details of 214,873,738 chess games played between January 1999 and January 2016 from the FICS Games Database website [12]. We then extracted each game's metadata, which included the users who played the game and the time the game was played. Using these details, we constructed a complex network $G_t^C := < V_t^C, E_t^C >$, in which the vertices V_t^c is a set of all FICS users in our dataset, and $e := (u,v) \in E_t^C$ is the list of all edges between the FICS users $u \in V_t^c$ and $v \in V_t^c$, where u and v played at least one game on FICS during the t weeks since the first game in our dataset. To study how the chess games network evolves over time, we constructed the network's graph every 4 weeks over a period of 18 years.

3.1.3 The WikiTree Marriage Network

We constructed a large social network using online genealogical records obtained from the WikiTree website [13]. WikiTree is an online genealogical website, created by Chris Whitten in 2008, with a mission to create a single worldwide family tree that will make genealogy free and accessible. The website contains over 13 million profile pages of people who lived in the previous centuries, and many of the profiles contain specific details about each individual, including full name, gender, date of birth, children's profiles, and spouses' profiles. To keep WikiTree's data integrity, only invited users can contribute, and contributors must agree to follow an honor code which specifies how they should treat openness, accuracy, mistakes, and giving credit. Moreover, many profiles reference the source of the data presented in the profile. Additionally, most profiles have a manager who has responsibility for WikiTree profiles [27], and each profile has its own "Trusted List" of people who have access to modify the profile, making the information in many profiles only editable to a limited number of people [28].

In 2015, Fire and Elovici [29] showed that it possible to utilize WikiTree data to create a large-scale social network that can be used to better understand lifespan patterns in human population. Similar to Fire and Elovici's study, in this study we utilized WikiTree data, which was downloaded in April 2016 and includes 1,964,331 marriage records of people whose birth years were between 1400 and 2010, to construct the marriage social network. Namely, we constructed a WikiTree marriage network graph at year y to be: $G_y^W := < V_y^W, E_y^W >$, where V_y^W is a set of people who, according to WikiTree's records, were born after 1400 and were married at least once before or during the year y, and each link, $e := (u,v) \in E_y^W$, is between two individuals, $u, v \in V_y^W$, who got married before or during the year y.

3.1.4 The Co-authorship Networks

The Microsoft Academic Graph is a large-scale dataset which contains scientific publication records of 126 million papers, along with citation relationships between those publications, as well as relationships between authors, institutions, journals, conferences, and fields of study [10]. The dataset also contains field-of-study hierarchy with four levels, L0 to L3, where L0 is the highest level, such as a research field of Computer Science, and L3 is the lowest level, such as a research field of Decision Tree [30].

In this study, using field-of-study hierarchy, we selected all the research fields within level L3 which contained at least 1,000 publications. For each selected research field, we constructed the field's co-authorship social network over time. Namely, let R be a selected research field, and let y be a year between the time of the first and last publication in R. We define the undirected co-authorship social network of R at y to be $G_y^{R_{co}} := < V_y^{R_{co}}, E_y^{R_{co}} >$, where $V_y^{R_{co}}$ is a set of authors who published a paper in R with a publication year before or including y. In addition, each link in the dataset $e := (u,v) \in E_y^{R_{co}}$ is between two authors, $u \in V_y^{R_{co}}$ and $v \in V_y^{R_{co}}$, who collaborated on a publication in field R with a publication year before or including y. Using the Microsoft Academic Graph dataset which was published for the KDD Cup 2016, we succeeded in constructing the social networks of 9,005 research fields over a period of 215 years. Overall, we created 770,845 co-authorship graphs.

It is important to notice that even though the co-authorship network graphs are undirected, for features calculations we treated these graphs as directed graphs where each undirected link $e := (u,v) \in E_y^{R_{co}}$ was transformed into two directed links between u and v, and also between v and u.

3.1.5 The Citation Networks

Similar to the construction of the co-authorship networks described above, we utilized the Microsoft Academic Graph to construct the citation networks within the lowest field-of-study hierarchy category of L3. Namely, let R be a selected research field, and let y be a year between the time of the first and last publication in R. We define the directed citation network of R at y to be $G_y^{R_{ci}} =< V_y^{R_{ci}}, E_y^{R_{ci}} >$, where $V_y^{R_{ci}}$ is a set of papers that were published in R with a publication year before or including y. In addition, each directed link in the dataset $e := (u,v) \in E_y^{R_{ci}}$ is between two papers $u \in V_y^{R_{ci}}$ and $v \in V_y^{R_{ci}}$, in which paper u cited paper v. Overall, we

[5] www.freechess.org/

constructed the citation networks of 8,996 research fields, which include 769,793 directed graphs.

3.1.6 The Bitcoin Transaction Network

Bitcoin is a cryptocurrency and a large-scale payment system, in which all the transactions are publicly accessible [31]. In this study, we used the Bitcoin Transaction Network Dataset published in 2013 by Ivan Brugere [14]. The dataset includes over 37.4 million transactions, from January 2009 to April 2013, between public-key "addresses," from which we created a directed network with over 6.3 million vertices and 16.3 million links over a period of 222 weeks. Namely, we defined the Bitcoin graph at time t to be $G_t^B :=< V_t^B, E_t^B >$, where V_t^B is a set of public-key addresses which perform their first transaction before time t, and $e := (u,v) \in E_t^B$ between two public-key addresses, $u \in V_t^B$ and $v \in V_t^B$, where according to the dataset a payment transaction was performed from u to v.

3.2 Analyzing Temporal Dynamics of Networks

3.2.1 Calculating Network Features

Throughout this study, we calculated various networks' features and analyzed how these features change over time. In this section, we provide formal definitions of these features. First, we define the graph of network n at time t to be $G_t^n :=< V_t^n, E_t^n >$. Then, we present the following network features:

- *Vertices number* - the number of vertices in the network at time t, defined as $|V_t^n|$.
- *Edges number* - the number of edges in the network at time t, defined as $|E_t^n|$.
- *Density* - the network's density at time t, defined as $D_t^n = \frac{|E_t^n|}{|V_t^n| \cdot (|V_t^n|-1)}$
- *Network active time* - a network's active time (denoted as t_{max}^n), defined as the amount of time between the times the first and last vertices joined the network.
- *Average clustering coefficient* - the coefficient that measures the level to which vertices in a graph tend to cluster together [32], defined at time t (denoted by CC_t^n) to be G_t^n's average clustering coefficient.
- *Average shortest path* - the network's average shortest path at time t (denoted by Avg. SP_t^n), defined as G_t^n's average shortest path.
- *Vertex degree* - for a vertex v in network n, we define the vertex degree at time t as $d_t^n(v) = |\{u|(u,v) \in E_t^n \text{ or } (v,u) \in E_t^n\}|$, i.e., the number of vertices at n that connect to v at time t
- *K-Stars set* - using the degree definition, we define the *K-Stars* set of n at time t (denoted by $Stars_t^n(k)$) to be the set of k vertices in n with the highest degree at time t. Namely,

$$\begin{aligned} Stars_t^n(k) \quad := \quad & \{v_1,\ldots,v_k | d_t^n(v_i) \geq d_t^n(v_j) \\ & \forall v_i \in \{v_1,\ldots,v_k\}, \forall v_j \notin \{v_1,\ldots,v_k\} \\ & \text{, and } v_i, v_j \in V_t^n\}. \end{aligned}$$

- *K-Stars-Vector* - using the *K-Stars set*, we can define a network's *K-Stars-Vector* over a monotonous time series $t_0, t_1, \ldots, t_m$ (denoted as $v_{k,n}^*$) to be the vector of size of m in which each ith element, i.e., $(v_{k,n}^*)_i$, represents the number of new emerging network stars at time t_{i+1}. Namely, let there be a network n which was active for time t_{max}^n and let there be a monotonous time series $t_0, t_1, \ldots, t_m, \forall t_i < t_{i+1}$, in which $t_m \leq t_{max}^n$. We define *K-Stars-Vector* over t_i to be

$$v_{k,n}^* = (|Stars_{t_i}^n(k) - \bigcup_{j=0}^{j=i-1} Stars_{t_j}^n(k)|)_{i=1}^m.$$

  In creating the *K-Stars-Vectors* for the networks analyzed in this study, we used a time series $t_0, t_1, \ldots, t_m$, in which $t_0 = 0$ and $t_m = t_{max}^n$, and the time difference between t_i and t_{i+1} was set to one year for the co-authorship and citation networks, and typically set to 4 weeks for the subreddit networks (in cases where the overall time did not divide evenly into 4-week intervals, the final interval was less than 4 weeks).
- *K-Stars-Number* – using the *K-Stars-Vector*, we can define the number of emerging stars in a network to be the number of unique vertices that, at a certain time, were among the top K vertices with the highest degree in the network. Namely, for a network n which was active for a time t_{max}^n, we define the *K-Stars-Number* of n, over time series $t_0, t_1, \ldots, t_m$, to be the sum of the *K-Stars-Vector* values

$$|v_{k,n}^*| = ||v_{k,n}^*||_1 := \sum_{i=1}^m (v_{k,n}^*)_i.$$

3.2.2 Vertices' Join-Time Difference

Similar to Price's observation [15] that new papers tend to be cited more than older papers, we noticed that in all six examined networks, vertices tended to connect to vertices that joined the network at a similar time: (a) Reddit users tend to be more engaged with other users who joined the network at a similar time, and to be less engaged with users who became active either a long time before or after they did; (b) online chess players tend to play more with other players who played their first game at a similar time, and to play less frequently with those who played their first FICS game either a long time before or after they did; (c) in the WikiTree dataset, people tend to marry others who are about the same age, and to marry less often those with whom there is a larger age gap; (d) researchers tend to collaborate more with other

researchers who published their first paper about the same year, and to collaborate less with researchers who published their first paper a considerable time before or after; (e) papers tend to cite papers more frequently that were published about the same time, and to cite older papers less frequently; and (f) Bitcoin transactions tend to occur more often between public-key addresses that became active about the same time, and less often between addresses that became active either a long time before or after.

To validate our observations, for each edge e in the Reddit, chess, WikiTree, co-authorship, citation, and Bitcoin networks, we calculated the join-time difference between each edge's vertices for all the edges in our dataset. We used regression analysis to calculate, across all networks, the probability of a vertex v connecting to a vertex u as the function of the time difference between the join times of u and v.

3.2.3 Network Join-Rate-Curves

Out of the six datasets we utilized, our study focused on the three datasets – Reddit, co-authorship, and citation – that had defined communities within the overall network, so that we could effectively analyze the evolution of their subnetwork structures. We defined the Join-Rate-Curve of a network n (denoted as JRC_n) to be the ratio of the number of vertices at time t and the maximal number of vertices in the network. Namely, let n be a network that was active for a time period of t^n_{max}; then, for $t \in [0, t^n_{max}]$, we define $JRC_n(t) \rightarrow [0,1]$ as:

$$JRC_n(t) := \frac{|V^t_n|}{|V^{t^n_{max}}_n|},$$

where $JRC_n(0)$ and $JRC_n(t^n_{max})$ are defined to always be equal to 0 and 1, respectively.

To create the JRCs for the selected networks, for each network n we calculated the JRC_n values using 4-week intervals for the subreddit networks, and using 1-year intervals for the co-authorship and citation networks of selected research fields. By using these intervals, the number of samples of the JRCs for the subreddit networks ranged from 1 to 141, with a median value of 51; and for both the co-authorship and citation networks ranged from 9 to 217, with a median value of 76.

To better understand the various types of JRCs that we created, we utilized CurveExpert software [33] to match several selected JRCs with their best-fit functions using regression analysis. To avoid overfitting, we selected the best-match function with relatively low degrees of freedom. Next, to verify that the selected function actually matched most of the JRCs, we used the python-fit package[6] to fit the selected best-match function on all 38,129 JRCs. Then, to better understand the various types of JRCs, we drew all the JRCs and ordered them according to the networks' vibrancies (see Section 3.2.4) in descending order. Lastly, we manually examined the various figure collections and scrutinized the anomalous JRCs that did not fit the selected regression function in terms of R^2

[6] https://pypi.python.org/pypi/python-fit/1.0.0

3.2.4 Network Vibrancy

We also observed differences in topological properties among networks with different growth rates. To better understand these differences, we defined the vibrancy of network n to be one minus the average value of the JRC_n function:

$$vibrancy(n) := 1 - \int_0^{t^n_{max}} \frac{JRC_n(t)}{t^n_{max}}.$$

The network vibrancy values range between 0 and 1, where vibrant, fast-growing networks usually have vibrancy values near 1, while slow-growing networks have vibrancy values near 0. To analyze the influence of different growth rates on the network topological properties, we calculated the Spearman correlations among the network topological properties (presented in Section 3.2.1) and the network vibrancies.

3.2.5 The Emergence of New Network Stars

One of the main goals of this study was to better understand how new network stars emerge. To achieve this, we analyzed the Spearman correlations between the frequencies at which network stars emerge (i.e., *K-Stars-Number*, for $K = 1,5$) and other network properties.

Moreover, we investigated in which stage of the network's life new stars are more likely to emerge by performing the following: First, for each type of network, we divided the network into two sets according to the network growth speed: fast-growing networks with vibrancies higher than 0.5 ($v_b > 0.5$), and slow-growing networks with vibrancies lower than 0.5 ($v_b < 0.5$). Next, for each set, we calculated the average number of network stars which emerged in each time slice, using time slices that were common for at least w networks. Namely, letting N be a set of networks, we define the $w-maximaltime$ to be the maximal time for which at least w networks in the set were active:

$$t^N_{w,max} := \max(\{t^n_{max}, n \in N \quad | \quad \exists n_1, n_2, \ldots, n_w \in N, \forall i \in [1,w]\ t^n_{max} \leq t^{n_i}_{max}\}).$$

Next, for a monotonous time series $t_0, t_1, \ldots, t_m$, where $t_m = t^n_{w,max}$, and for each time stamp $t_i \in \{t_1, \ldots, t_m\}$, we measured the average number of new stars that emerged between times t_{i-1} and t_i by calculating the *K-Stars-Vector* of each network $n \in N$ over $t_0, t_1, \ldots, t_m$ and calculating the average number of emerging stars for each t_i between t_1 and t_m. Namely, we define the following vector:

$$TotalStarsVector^N_k := (\sum_{\{n \in N | t_i \leq t^n_{max}\}} (v^*_{k,n})_i)^m_{i=1},$$

where $TotalStarsVector^N_k$ is an m-length vector, in which each ith element is the sum of the number of emerging stars at time t_i, across all networks in $n \in N$ with an active time of at least $t_i (t_i \leq t^n_{max})$. Then, using $TotalStarsVector^N_k$, we define the $AvgStarsVector^N_k$ by dividing each ith element in $TotalStarsVector^N_k$ by the number of networks that were active for a time of at least t_i:

$$AvgStarsVector^N_k := (\frac{TotalStarsVector^{N_i}_k}{||\{n \in N | t^n_{max} \geq t_i\}||})^m_{i=1}$$

Additionally, to reduce the influence of networks with frequently emerging stars on the $AvgStarsVector_k^N$, we also define the $NormAvgStarsVector_k^N$ as a vector with m elements in which each ith element is the average normalized value of the number of emerging stars at time t_i across all networks in $n \in N$:

$$NormAvgStarsVector_k^N := (\frac{(NormTotalStarsVector_k^N)_i}{||n \in N | t_{max}^n \geq t_i\}||})i = 1^m,$$

where $NormTotalStarsVector_k^N$ is defined as:

$$NormTotalStarsVector_k^N := (\sum_{\{n \in N | t_i \leq t_m ax^n\}} \frac{(v_{k,n}^*)_i}{|v_{k,n}^*|})_{i=1}^m.$$

In this study, we calculated $AvgStarsVector_k^N$ and $NormTotalStarsVector_k^N$, for $k = 1, 5$.

4. Results

4.1 Vertices' Join-Time Difference

By analyzing the vertices of over 8.3 billion edges and using probability calculations and regression analysis, we discovered that across all networks, the probability of a vertex v connecting to a vertex u decreases sharply, typically in an exponential decline rate, as the time difference between the join times of u and v increases (Figure 1).[7]

4.2 Network Join-Rate-Curves

As described in Section 3.2.2, to better understand the different rates in which vertices join networks, we examined and analyzed over 38,000 JRCs. We discovered that in most cases, the best fit was a high-degree polynomial function. To avoid overfitting, we used the CurveExpert software [33] and python-fit package to find the polynomial function that was a best fit for the majority of JRCs and still had a relatively low degree. We discovered that among all the subreddit networks, 18,558 (92.2%) and 14,505 (72.06%) of the JRCs matched quartic functions ($q(x) := a + bX + cX^2 + dX^3 + eX^4$) with $R^2 \geq 0.95$ and $R^2 \geq 0.99$, respectively. In addition, among all the research field co-authorship networks, 8,508 (94.49%) and 5,465 (60.68%) of JRCs matched quartic functions with $R^2 \geq 0.95$ and $R^2 \geq 0.99$, respectively. Furthermore, among all the research field citation networks, 8,568 (95.2%) and 5,910 (65.7%) of JRCs matched a quartic function with $R^2 \geq 0.95$ and $R^2 \geq 0.99$, respectively.

After observing that the vast majority of JRCs match quartic functions, our next goal was to better understand which type of quartic function the JRCs frequently match. We achieved this by drawing all 38,129 JRCs and ordering them according to the networks' vibrancies in descending order (see Figure Collections S1, S2, and S3). Using this methodology, we observed five common JRC patterns – polynomial, sublinear, linear, superlinear, and sigmoidal (see Figure 2). Additionally, by analyzing the JRCs that did not match quartic functions, we identified a sixth type of JRC which was influenced by external events, such as the HalloweenCostume subreddit JRC that gains popularity near Halloween each year, or the JRC Quasicrystal research field citation network that demonstrates an interesting growth pattern, probably due to a paradigm shift in the field.

4.3 Network Vibrancy

Using correlation calculations, we discovered various correlations between the vibrancies and other network characteristics (see Figures 3 and 4). Primarily, we discovered the following correlations: (a) medium-to-high positive correlations ($r_s = 0.78, 0.73, 0.5$) between the networks' vibrancies and the duration in which the networks were active; (b) small-to-medium positive correlations ($r_s = 0.28, 0.3, 0.38$) between the networks' vibrancies and the number of vertices; (c) small negative correlations ($r_s = -0.23, -0.35, -0.33$) between the networks' vibrancies and densities; and (d) small negative correlations ($r_s = -0.12, -0.21, -0.17$) between the networks' vibrancies and the average clustering-coefficients. According to these correlation results, we can discern that networks with high vibrancy tend to be active longer, have more vertices, be less dense, and cluster less than networks with low vibrancy.

It is worth mentioning that most of the studied co-authorship and citation networks presented relatively high vibrancy values, which usually indicates fast-growing networks, while the subreddit networks presented both high and low vibrancy values (see Figure 3). We believe this is a result of our research field selection process, in that we chose only successful research fields with over 1,000 published papers (see Sections 3.1.4 and 3.1.5).

4.4 Emergence of Network Stars

We discovered correlations between the vibrancy of networks and the changes in their most-connected vertices, i.e., their stars. By measuring how a list of top-5 network stars changed over time, we found medium-to-high positive correlations ($r_s = 0.44, 0.44, 0.73$) between the networks' vibrancies and the total number of changes in the top-5 stars. Additionally, there were medium-to-high positive correlations ($r_s = 0.56, 0.71, 0.7$) between the duration the networks were active and the top-5 network stars. Moreover, across all networks, we analyzed how the number of emerging stars changed over time (see Section 3.2.5). For networks with low vibrancy, most stars emerged a short time after the network became active and kept their place, while for networks with high vibrancy, stars emerged at any time (see Figures 5 and 6). These results indicate that stars tend to emerge in networks that are growing rapidly.

5. The TPA Network-Generation Model

In many complex networks the rich tend to get richer, known as the preferential-attachment process [6]. Incorporating this

[7] The figures throughout this paper were created with high resolutions, which makes it possible to review each figure's details by zooming into the figure.

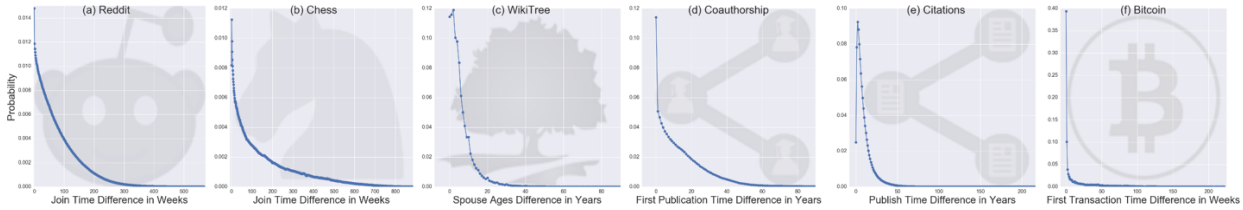

Figure 1. The probability of two vertices connecting, as a function of the time the first joined the network. In all six real-world networks, as the join-time difference increases, the probability of vertices connecting decreases sharply, estimated by the following functions: (a) $0.012e^{\frac{-t}{88.38}}$ is the probability of a Reddit user replying to another user, where t is the time difference in weeks; (b) $0.0064e^{\frac{-t}{145.28}}$ is the probability of two chess players playing against each other, where t is the time difference in weeks; (c) $0.138e^{\frac{-t}{7.01}}$ is the probability of two people getting married, where t is their age difference in years; (d) $\frac{0.179-0.008t^{0.642}}{1.614+t^{0.642}}$ is probability of two authors coauthoring a paper, where t is the time difference in years; (e) $\frac{0.049+0.004t}{1-0.221t+0.04t^2}$ is the probability of one paper citing another paper, where t is the years between the papers' publications; and (f) $0.391e^{\frac{-t}{0.799}}$ is the probability of Bitcoin transactions between two accounts, where t is the time difference in weeks.

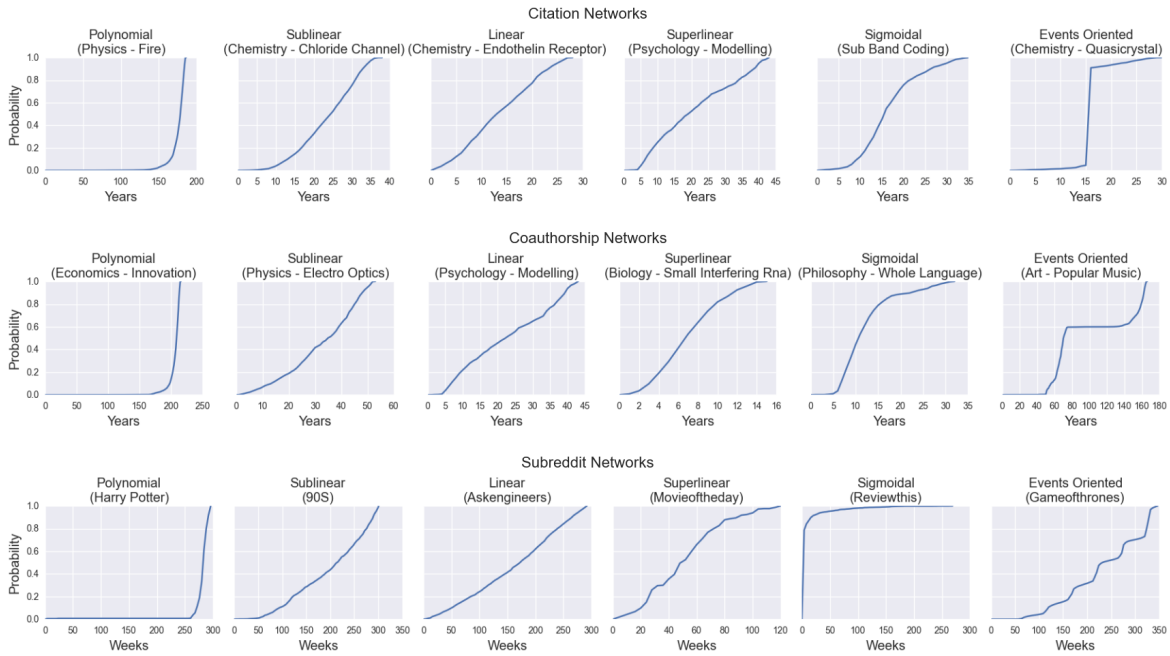

Figure 2. Common Join-Rate-Curve patterns. We observed five types of common JRC patterns – polynomial, sublinear, linear, superlinear, and sigmoidal. Additionally, we identified a sixth type of JRC with distinct growth patterns that are greatly affected by events, such as the quasicrystal citations network's JRC (right column) which demonstrates the field's sudden increase in popularity. These various growth patterns result in different network topological properties.

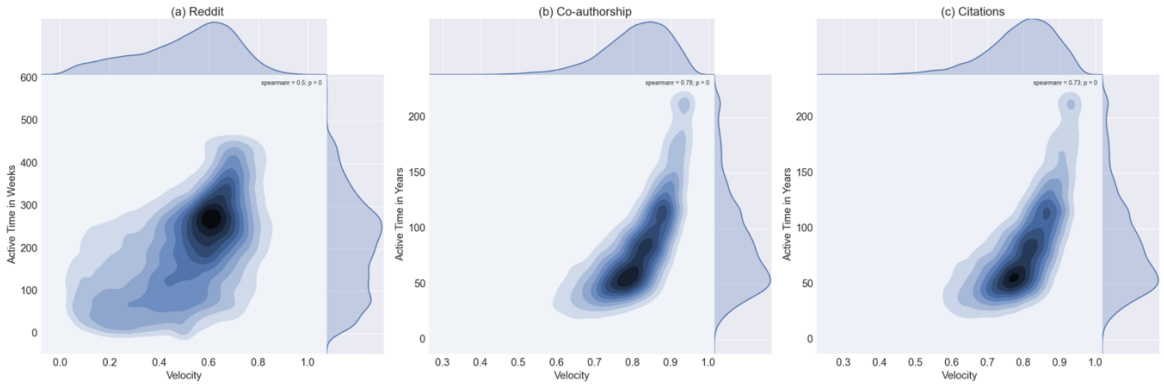

Figure 3. Joint distributions of network vibrancy (V_b^n) and active time (T_{max}^n). Networks with relative high vibrancy tend to be active longer. Additionally, note that the subreddit networks have quite diverse vibrancy values, while the co-authorship and citation networks have mostly high vibrancy values.

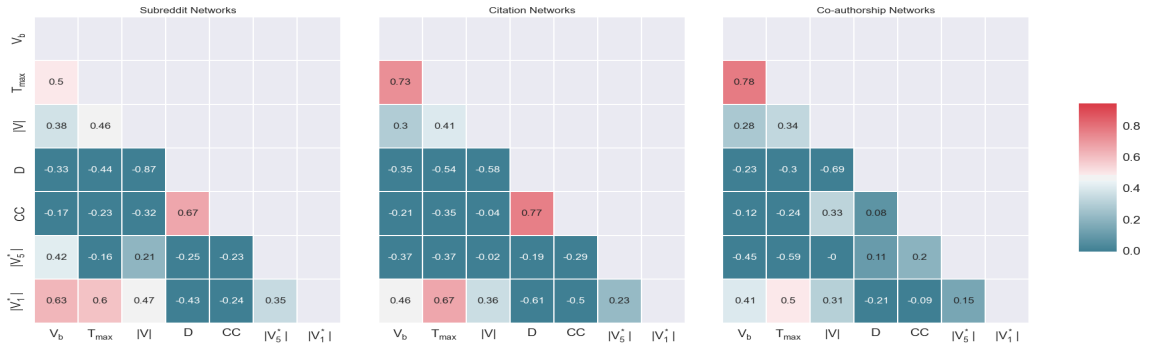

Figure 4. Correlation matrices. Correlations exist among these network features: V_b, vibrancy; T, duration the network was active; $|V|$, number of vertices; D, density; CC, average clustering coefficient; $|V_5^*|$, total number of changes over time in the top-5 network stars; and $|V_1^*|$, total number of changes over time in the top network star (i.e., how many times the most-linked vertex was changed). For these three types of networks there are high positive correlations between V_b and T, as well as medium-to-high positive correlations between V_b and both V_1^* and V_5^*, as well as between T and both V_1^* and V_5^*.

tenet into the above observations, we can obtain a more complete picture of how networks evolve and how network stars emerge. For example, consider an online social network growing at a very fast rate, i.e., with vibrancy close to 1. As a result of the preferential-attachment process, there will quickly be several high-degree users. However, in line with our first and second observations, as the network continues growing rapidly, these initial highly connected users will gain fewer connections as new-generation users will mainly connect among themselves. According to the preferential-attachment process, new *local* generation stars will emerge among the new-generation users.

Since the network is growing quickly, new users outnumber old users. Therefore, new-generation stars will eventually have more connections than old stars and will become *global* stars. This process will repeat itself as long as the network keeps growing quickly. However, if the growth rate abruptly declines, the network will become more clustered and dense, resulting in fewer emerging local network stars that later become global stars. Inspired by the above observations, we developed the TPA model, which mimics the behavior of real complex networks. This model generalizes the well-known Barabási-Albert network generation model (denoted BA model) [6] by incorporating the role of time. Instead of adding only one vertex in each iteration, the TPA model supports the rate in which vertices actually join the network, as well as the number of links each vertex establishes when it joins. Moreover, the TPA model includes as input the probability that each vertex will connect to other vertices with the same or different join-time. The presented TPA model produces arbitrary-sized, random scale-free networks with relatively high clustering coefficients, which are sensitive to vertex arrival times and to the network's vibrancy.

In the following subsections, we will describe in the detail the TPA model, and the evaluate the properties of networks generated by TPA model's properties alongside with similar size networks which were generated by the classic BA and Small-World network models. Moreover, an implementation

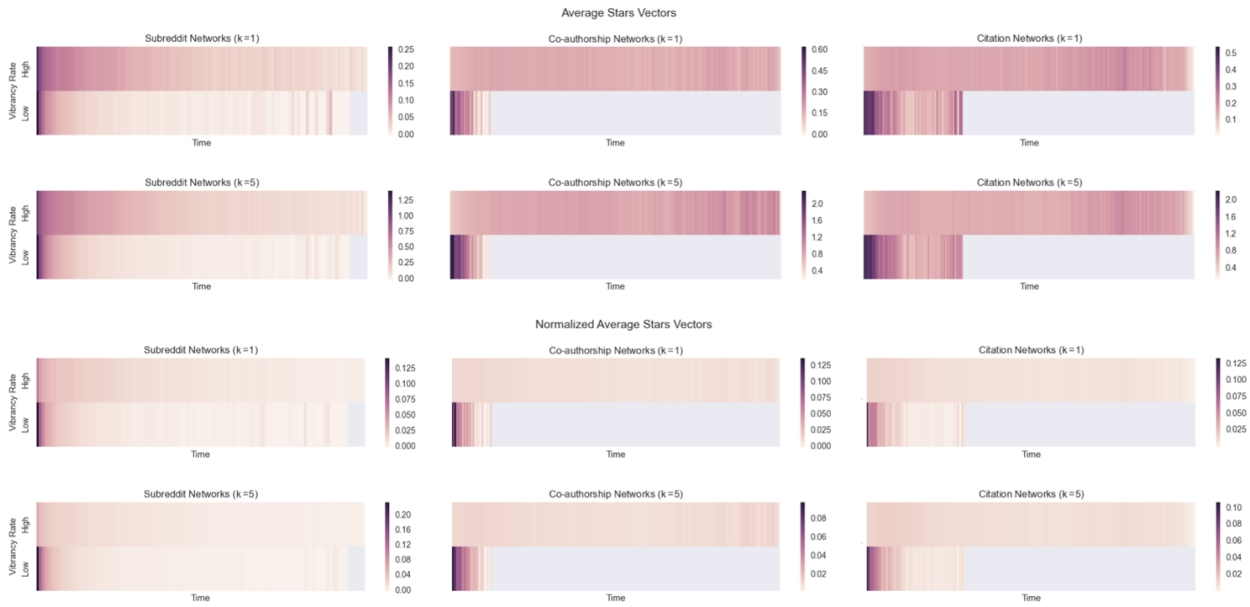

Figure 5. New network star emergence over time. The tendency of a new star to emerge in a network with low vibrancy is much greater in the beginning than after the network matures. Additionally, for a network with high vibrancy, new stars frequently emerge at the very beginning of the network's life and tend to emerge in similar probabilities afterwards.

of the model, including code examples, can be found at the project's website (see Section 8).

5.1 TPA Model Algorithm

An overview of the TPA model algorithm is presented in Algorithm 1 and Figure 7. The TPA model receives as input three parameters: first, the number of edges (denoted m) to attach a new vertex to existing vertices; second, an integers list (denoted l) with the number of vertices to add to the graph in each iteration; and third, a function (denoted $f : N \rightarrow [0,1]$) that, given a time difference value, returns the relative probability of an edge existing across two time groups. The algorithm starts by creating an empty undirected graph (line 1) and an empty time group list (line 2).

Then, for each positive integer $l[i]$ in l, the algorithm does the following:

- Creates new $l[i]$ vertices with the time group set to i (lines 5-6);
- Adds the new vertices to the graph (line 7);
- Adds i to the TimeGroupsList (line 8);
- Connects each new added vertex to the other m vertices using the AddRandomEdges procedure (line 10).

The *AddRandomEdges* procedure (lines 12-18) is the core of the model. The procedure receives as input five parameters: a graph (g), a vertex (v), the number of edges (m), a probability time difference function (f), and a list of existing time groups ($TimeGroupsList$). The *AddRandomEdges* procedure connects v to m other vertices in the graph using the following routine:

Algorithm 1 The Temporal Preferential Attachment Model Algorithm Overview

```
 1: procedure GenerateRandomGraph
Require: m, l, f
 2:     g ← empty undirected graph
 3:     TimeGroupsList ← empty list
 4:     for i = 0 to l.length do
 5:         NewVerticesList ← list of l[i] new vertices
 6:         AddNewVerticesTimeGroup(NewVList,i)
 7:         AddVerticesToGraph(g NewVList)
 8:         AddToList(TimeGroupsList, i)
 9:         for v in NewVerticesList do
10:             AddRandomEdges(g,v,m,f)
11:
12: procedure AddRandomEdges
Require: g,v,m,f
13:     for j = 0 to m do
14:         repeat
15:             r ← SelectFromList(TimeGroupsList,f)
16:             u ← GetVertexInGroupByDegree(g,r)
17:         until (v,u) ∉ g
18:         AddEdge(g,v,u)
```

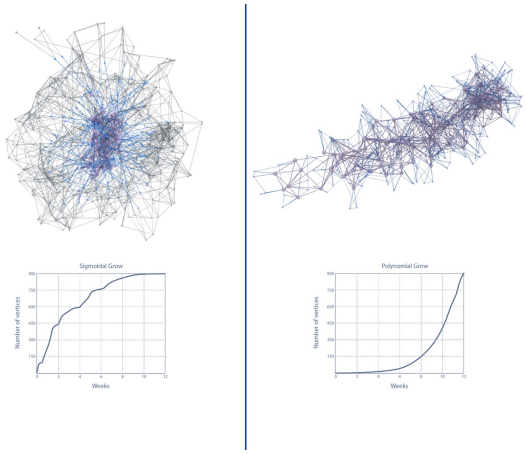

Figure 6. Star emergence in fast- and slowgrowing networks. By analyzing the network evolution process, we observed that in slow-growing networks, such as the one on the left, most stars (pink vertices) emerged a short time after the network became active and kept their place, while for fast growing networks, such as the one on the right, stars emerged at any time (see Video S1). The graphs above are for illustrative purposes.

1. It randomly selects from *TimeGroupsList* a time group (denoted r) where the probability of selecting each time group is given by f (line 1), where given $t_1, t_2, \ldots, t_n$ time groups, the actual probability of an edge being created between two time groups with a time difference of $d \leq n$ is equal to $\frac{f(d)}{\sum_1^n f(t_i)}$.
2. Similar to the BA model, the procedure selects one vertex (u) among all the vertices that are in the selected time group r, where vertices with higher degree have higher likelihood of being selected (line 16). In case the edge (u, v) already exists in the graph, then the selection process of u is repeated until a new u in the graph is created.[8]

To illustrate our TPA model algorithm, we can create a random graph using the following input parameters: $m = 3$, $l = (100, 200, 400)$, and $f(t) = 2^{-1-t}$. We start running the model with an empty graph. In the first iteration, we add 100 ($l[0]$) new vertices to the graph, and each new vertex has a time group value of 0. In this iteration there are not any other time groups. Therefore, the 100 new vertices will create only 300 $(100 \cdot 3)$ edges among themselves in the following manner: each vertex will select 3 other vertices in the group, and, similar to the BA model, vertices with higher degree will have higher probability of being selected, i.e., the richer vertices will have a higher probability of becoming richer.

In the second iteration, the model will insert 200 $(l[1])$ new vertices, which will form 600 $(200 \cdot 3)$ new edges. However, this time we have two time groups: (a) a time group of 1 (with time difference 0), which contains all the 200 new vertices, and according to the time difference probability function, the probability of each new vertex establishing a connection to this group is $f(0) = 2^{-1-0} = 0.5$; and (b) a time group of 0 (with time difference of 1), which contains the previous 100 vertices, with a probability of $f(0) = 2^{-1-1} = 0.25$ of connecting to vertices in this time group. According to these parameters, we can observe that the probability ratio of the two time groups is 2 to 1. Therefore, we can use this ratio to estimate that out of the 600 edges of the second iteration, about 400 edges will be formed among the vertices of time group 1, and about 200 edges will be formed among the vertices of time group 1 and time group 0, where each edge has a higher probability of connecting vertices with higher degree. A detailed implemented TPA model in Python can be found in the paper's website

Lastly, in the third iteration, our model will insert an additional 400 $(l[2])$ new vertices to the graph with a time group value of 2. These vertices will formulate $1,200$ $(400 \cdot 3)$ edges, of which about 686 will be among the vertices of time group 2; about 343 edges will be among the vertices of time groups 2 and 1 (time difference of 1); and about 171 edges will be among the vertices of time groups 2 and 0 (time difference of 2). Overall, the TPA model will have constructed a graph with 700 vertices and $2,100$ edges.

5.2 TPA Model Evaluation

To empirically evaluate the TPA model, we created various random networks using various input parameters: The vertices number was set to three different sizes: $700, 6,200$, and $12,350$. The edge number, parameter m, was set to 3, creating networks with about $2,100, 18,600$, and $37,050$ edges. We used linear, polynomial, and sigmoidal vertex growth rates. For the linear growth rate, we added 10 new vertices in each iteration. For the polynomial growth rate, we used the sequence of $5, 20, 45, ..., 5x^2$, with a maximal x value of $8, 16$, and 20 for creating networks with $2,100, 18,600$, and $37,050$ edges, respectively. For the sigmoidal growth rate, we used the same growth sequence as used in polynomial growth, only in reverse order. We used $f(t) = 2^{-1-t}$ and $f(t) = 0.8 \cdot 0.2^t$ functions as time difference functions (f), where $f(t) = 0.8 \cdot 0.2^t$ will create considerably more edges among all the vertices in the same time group than $f(t) = 2^{-1-t}$.

Overall, we assembled 18 different parameter settings for generating random networks. For each parameter setting, we utilized the TPA model to create 18 random networks. Subsequently, for each network, we calculated the network's average clustering coefficient, the maximal degree of vertex in the network, the network's average shortest path value, the *K-Stars-Number* for $k = 1, 5$ (denoted v_1^* and v_5^*), and the power-law function ($k^{-\gamma}$) that matched the degree distribution of the network. To reduce variance of the calculated features, we repeated the network construction process and feature calculations 10 times for each parameter setting and

[8] In the Python implementation of the TPA model (see Section 8), we limited the number of repeats to prevent cases where it is impossible to add new edges to v.

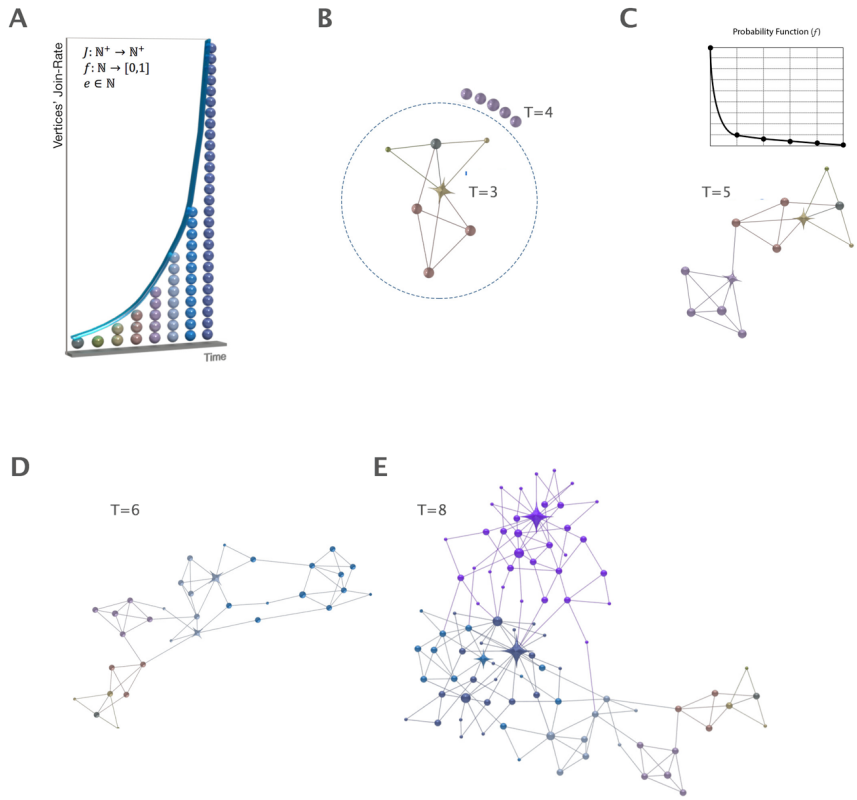

Figure 7. Temporal Preferential Attachment model. The TPA model generates scale-free complex networks in which new stars emerge over time using the following steps: **(A)** There are three input parameters: j, the vertices' join-rate over time; f, a monotonically decreasing function giving the probability of a vertex that arrives at time t_i connecting to other vertices that arrive at time t_j; and e, the number of edges each vertex establishes upon joining the network. **(B)** The model generates a random network as, in each time iteration, a group of new vertices joins the network together (each group as a different color). **(C)** Each vertex v establishes e new links, first by selecting the time group t_j to connect using the probability function f. Then, a random vertex that arrived at t_j is selected. The vertex selection process is very similar to the preferential-attachment process, i.e., a vertex is selected at random, where vertices with high degree have higher likelihood, proportional to the vertex degree, to be selected. Afterwards, a link is created between v and the selected vertex. **(D)** In each iteration new groups of vertices join the network. **(E)** As time passes, the degree of the new joined vertices suppresses the degree of the previously joined; i.e., new network stars, marked with a star shape, are emerging.

calculated the average value of each feature. The results of these calculations are presented in Table 2. Furthermore, for comparing the TPA model with other models, we used the BA model [6], the Watts-Strogatz model (denoted WS) [5], the Newman-Watts model (denoted NW) [**?**], the Holme and Kim model (denoted HK) [21], and the Forest Fire model (denoted FF) [**?**]. To generate random networks of similar sizes, where the p parameters in the WS and NS model was set to 0.1, in the HK model the probability of adding a triangle after adding a random edge was set to 0.2, and in the FF model the forward probability was set to 0.65 (see Table 2).[9]

[9]For the BA, WS, NW, and HK models, we utilized the graph generation code from the Networkx package (see https://networkx.github.io/documentation/latest/reference/generators.html. We used the IGraph package implementation for the FF model (see http://igraph.org/c/doc/igraph-Generators.html).

6. Discussion

By analyzing the results presented in Sections 4 and 5, the following can be noted:

First, as can be observed in Section 2, the field of complex networks is flourishing, with an ever-growing body of work and an increasing number of random network generation models. The massive corpora of networks created and released due to this study can greatly contribute to a better understanding of complex dynamic networks by identifying which existing network generation models reflect real-world complex networks. Moreover, the released corpora can help in creating better network models which more accurately mimic real-world network behavior. Network models must accurately reflect the real world so that we can gain a correct understanding of dynamic and complex systems.

Second, by examining the JRCs of over 38,000 networks,

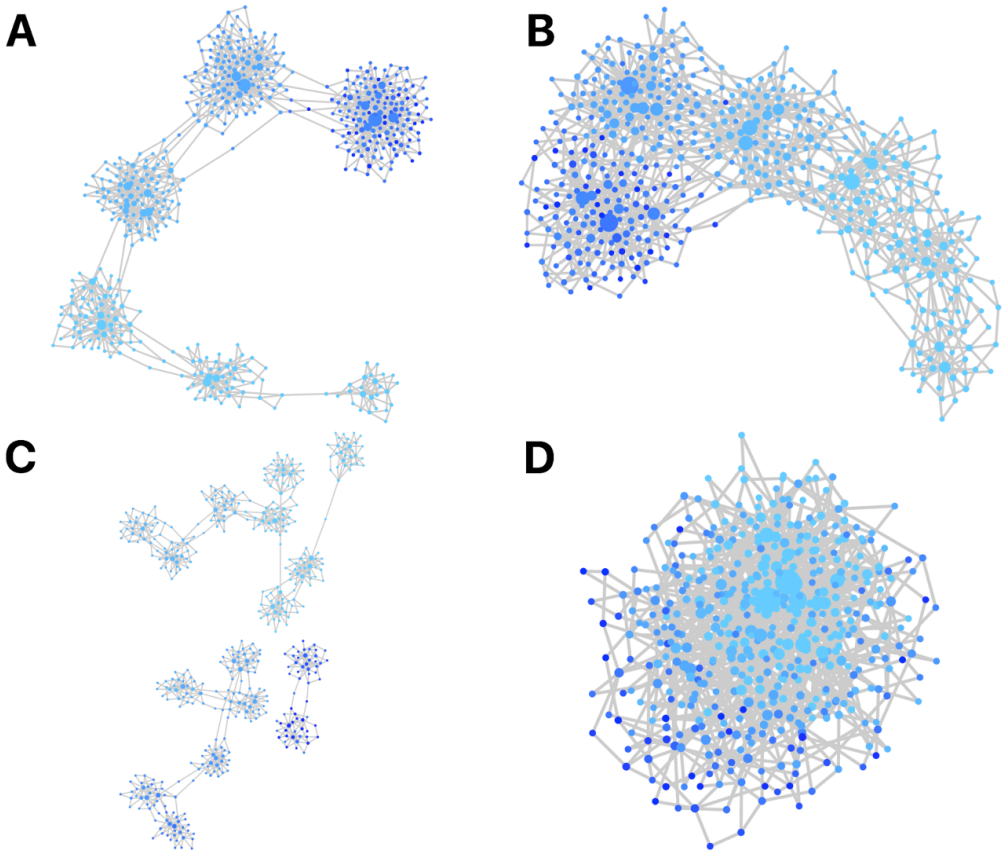

Figure 8. Networks with various topological structures created by the TPA model. Networks **A** and **B** were constructed using a fast growth rate and setting the time difference functions to create 98% and 86% of links in the same time group, respectively. Network **C** was created using constant growth by adding 30 vertices to the network in each iteration and setting the time difference functions to create 95% of the links in the same time group. Network **D** was created using a sigmoidal-like growth rate and setting the time difference functions to create 65% of the links in the same time group. In all four graphs the number of edges (m) was set to 2. Additionally, the color of the vertices in the graph represents the time in which the vertices were added to the networks; light blue vertices were added earlier than dark blue vertices. Also, the size of each vertex is proportional to degree of the vertex, i.e., larger vertices have higher degrees.

Table 2. Random Networks' Topological Properties.

Model	|V|	|E|	Rate	f	CC	d_{max}	Avg. SP	$\|v_1^*\|$	$\|v_5^*\|$	$k^{-\gamma}$
TPA	700	2100	Linear	2^{-1-t}	0.063	28.8	4.21	2	9.4	$k^{4.99}$
	700	2095	Poly.	2^{-1-t}	0.030	36	3.7	2.9	12.9	$k^{4.11}$
	700	2100	Sig.	2^{-1-t}	0.053	56.7	3.74	1.2	5.4	$k^{3.48}$
	700	2100	Linear	$0.8 \cdot 0.2^t$	0.150	28	5.95	2.6	13	$k^{4.15}$
	700	2095	Poly.	$0.8 \cdot 0.2^t$	0.080	46.4	3.97	4.5	20.9	$k^{3.62}$
	700	2100	Sig.	$0.8 \cdot 0.2^t$	0.097	56.1	4.17	1.4	6.8	$k^{3.41}$
	6200	18600	Linear	2^{-1-t}	0.046	32.6	13.65	2.1	16.3	$k^{6.98}$
	6200	18595	Poly.	2^{-1-t}	0.007	61.6	4.89	5.5	20.6	$k^{4.44}$
	6200	18600	Sig.	2^{-1-t}	0.012	119.5	4.92	1	5.4	$k^{3.6}$
	6200	18600	Linear	$0.8 \cdot 0.2^t$	0.141	31.6	31.38	4.9	21.3	$k^{3.22}$
	6200	18595	Poly.	$0.8 \cdot 0.2^t$	0.024	90.6	5.57	8.1	32.7	$k^{3.33}$
	6200	18600	Sig.	$0.8 \cdot 0.2^t$	0.029	113.8	5.83	1.2	7.6	$k^{3.46}$
	12350	37050	Linear	2^{-1-t}	0.045	33.7	24.1	3	16.5	$k^{8.5}$
	12350	37045	Poly.	2^{-1-t}	0.004	79.5	5.3	6	24.1	$k^{4.46}$
	12350	37050	Sig.	2^{-1-t}	0.007	154.6	5.36	1	6.2	$k^{3.67}$
	12350	37050	Linear	$0.8 \cdot 0.2^t$	0.138	33.9	60.3	5.1	22.8	$k^{3.18}$
	12350	37045	Poly.	$0.8 \cdot 0.2^t$	0.016	113.5	6.26	9.8	36.5	$k^{3.33}$
	12350	37050	Sig.	$0.8 \cdot 0.2^t$	0.020	163.6	6.51	1.1	7.9	$k^{3.42}$
BA	700	2091	-	-	0.039	80.5	3.37	2.3	10.8	$k^{3.12}$
	6200	18591	-	-	0.008	251.2	4.12	2.6	11.0	$k^{3.04}$
	12350	37041	-	-	0.004	341.7	4.35	2.9	10.5	$k^{3.06}$
WS	700	2100	-	-	0.444	8.9	5.76	-	-	-
	6200	18600	-	-	0.442	9.8	8.11	-	-	-
	12350	37050	-	-	0.443	10.1	8.85	-	-	-
NW	700	2306	-	-	0.510	9.8	5.56	-	-	-
	6200	20467	-	-	0.507	10.8	7.74	-	-	-
	12350	40763	-	-	0.507	11.2	8.44	-	-	-
HK	700	2090	-	-	0.136	86	3.37	2.5	11.3	$k^{3.17}$
	6200	18588	-	-	0.110	271.2	4.13	3.1	9.8	$k^{3.05}$
	12350	37038	-	-	0.107	456.8	4.33	2.8	10.1	$k^{3.07}$
FF	700	2067	-	-	0.538	294.5	3.49	-	-	$k^{3.48}$
	6200	21610	-	-	0.521	2354.6	3.75	-	-	$k^{3.35}$
	12350	43813	-	-	0.517	4689.5	3.78	-	-	$k^{3.38}$

we discovered six main common network growth patterns. We showed that there are differences between the structural properties of polynomial-growing networks and sigmoidal-growing networks. However, many questions such as *How prevalent is each type of growth pattern? Do other types of complex networks present similar types of growth patterns? Are there any other types of common growth patterns that may appear in other domains?* still remain open. We look forward to answering these questions in future research, when we look at the topologies of various complex networks, such as biological networks.

Third, as observed in our data, the time and rate in which vertices join a network have a crucial effect on the network's structure and dynamics. For example, as shown in Figures 3 and 4, fast-growing networks with high vibrancies and slow-growing networks with low vibrancies tend to present different topological features. Therefore, the time and rate in which networks evolve are two key factors that need to be included in understanding complex networks. In future studies, it would be interesting to analyze the effect of these two key factors on other network characteristics, such as the effect of different vibrancy values on the content of various Reddit communities. Additionally, we plan to explore how different input parameters influence the created network topology.

Fourth, network stars emerge differently in fast- and slow-growing networks (see Figures 5 and 6). In slow-growing networks, most stars emerge a short time after the network becomes active and keep their place, while in fast-growing networks, stars emerge at any time. Although it is not always possible to represent trends by a network's vertices, we believe these results help explain a wide range of changing trends, such as the emergence of fast-growing companies, the effect of disruptive technologies, and the rise of new mega-celebrities in online social networks.

Fifth, networks with low vibrancies typically have higher average clustering coefficient (*CC*) values than networks with high vibrancies (see Figure 4). This is confirmed by the networks generated by the TPA model, in which networks created by sigmoidal growth usually presented higher *CC* values than same-size networks created by polynomial growth (see Table 2).

Sixth, it is important to keep in mind that community networks can affect each other within a larger network [34]. For example, sudden growth in one research community can result in slowing growth in another research community. Moreover, as we demonstrated, the community growth patterns can also be affected from external events, such as a holiday. Even unconnected networks can influence each other. For example, even though Facebook, Twitter, and WhatsApp are different social platforms, they can considerably influence the network properties of each other. In future research, we plan to study the connections among various communities' growth patterns.

Seventh, the TPA model generates scale-free networks with similar degree distribution to networks created by the BA model, and with small average shortest path values, but with much higher CC values than the BA model (see Table 2). Furthermore, the TPA model can generate random networks with diverse topologies (Figure 8). While most vertices with high degree in the BA model likely joined the network in the first iterations, the TPA model's vertices joined the network in later iterations. This more accurately mimics a real-world network's evolution process and provides insight on how a newly added vertex can suddenly become popular, such as when a post becomes viral in social networks. In the future, we hope to analytically examine the TPA model and develop a similar model for creating directed and weighted networks.

Lastly, even though the TPA model is fairly simple, it can be used to answer a variety of questions, such as *Why do some platforms, like Facebook, flourish while other platforms, like MySpace, shrivel? Why do some viruses spread rapidly and others do not? How do specific individuals gain their power? When will a key person be replaced by a new star?* The TPA model currently offers a practical, straightforward way to simulate the evolution of complex dynamic systems using real data.

7. Conclusions

The field of data science has undergone many recent advances, and new algorithms, infrastructures, and techniques for data mining, data storage, data prediction, and data visualization have emerged [35, 36, 37, 38]. These tools make it feasible to gain new insights from vast quantities of data. In this study, we utilize data science tools to construct the largest publicly available network evolution corpora to date, in order to perform the first precise wide-scale analysis of the evolution of networks with various scales. Our study uses real data from actual networks.

We utilized the corpora to deeply examine the evolution process of networks and to understand how popularity shifts from one vertex to another over time. From our analysis, three key observations emerged: First, links are more likely to be created among vertices that join a network at a similar time. Second, the rate in which new vertices join a network is a central factor in molding a network's topology. Third, the emergence of network stars, i.e., high-degree vertices, is correlated with fast-growing networks. Based on these observations, we have developed a simple, random network generation model. Our Temporal Preferential Attachment (TPA) model more closely represents real-world data in fast-growing networks than previous models, many of which used a relatively small amount of data or only partial real data.

Moreover, the large corpus of networks created and released due to this study can greatly contribute to a better understanding of complex networks in general. We endorse the words of Albert-László Barabási: "If data of similar detail capturing the dynamics of processes taking place on networks were to emerge in the coming years, our imagination will be the only limitation to progress." [1] Much progress is being made in the field of complex networks, and our research emphasizes the value of using vast quantities of real data to create models that accurately represent the world around us. We must stay true to the real world to keep progressing in the right direction.

8. Data and Code Availability

One of the main goals of this study was to create the largest complex network evolution public dataset. Therefore, the Reddit, FICS Games, WikiTree, Microsoft Academic Graph, and Bitcoin Transaction datasets used to create the networks and graphs in this study are all open and public. The social network datasets and a considerable part of the study's code, including implementation of the TPA model and code tutorials, are available at the project's website which also gives researchers the ability to interactively explore and better understand the networks in this study's dataset (see Figure 9).

Acknowledgments

First and foremost, we would like to thank Jason Michael Baumgartner, Chris Whitten, Ivan Brugere, FICS Games Database team, and the Microsoft Academic Graph team for making their datasets available online. Additionally, we thank the AWS Cloud Credits for Research. We also thank the Washington Research Foundation Fund for Innovation in Data-Intensive Discovery, and the Moore/Sloan Data Science Environments Project at the University of Washington for supporting this study. Datasets, software implementations, code tutorials, and an interactive web interface for investigating the studied networks are available at https://dynamics.cs.washington.edu/

We also wish to thank Carol Teegarden for editing and proofreading this article to completion. Additionally, we thank Danilo Gutierrez for creating the project's video and Aaron Romm for contributing his voice to the video. We also thank Stephen Spencer for his IT expertise. Lastly, we wish to thank Dima Kagan and Hila Fire for their assistance during this research.

References

[1] Albert-László Barabási. Scale-free networks: a decade and beyond. *science*, 325(5939):412–413, 2009.

[2] Nataša Pržulj and Noël Malod-Dognin. Network analytics in the age of big data. *Science*, 353(6295):123–124, 2016.

[3] Albert-László Barabási. Linked: The new science of networks, 2003.

[4] Erzsébet Ravasz and Albert-László Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2):026112, 2003.

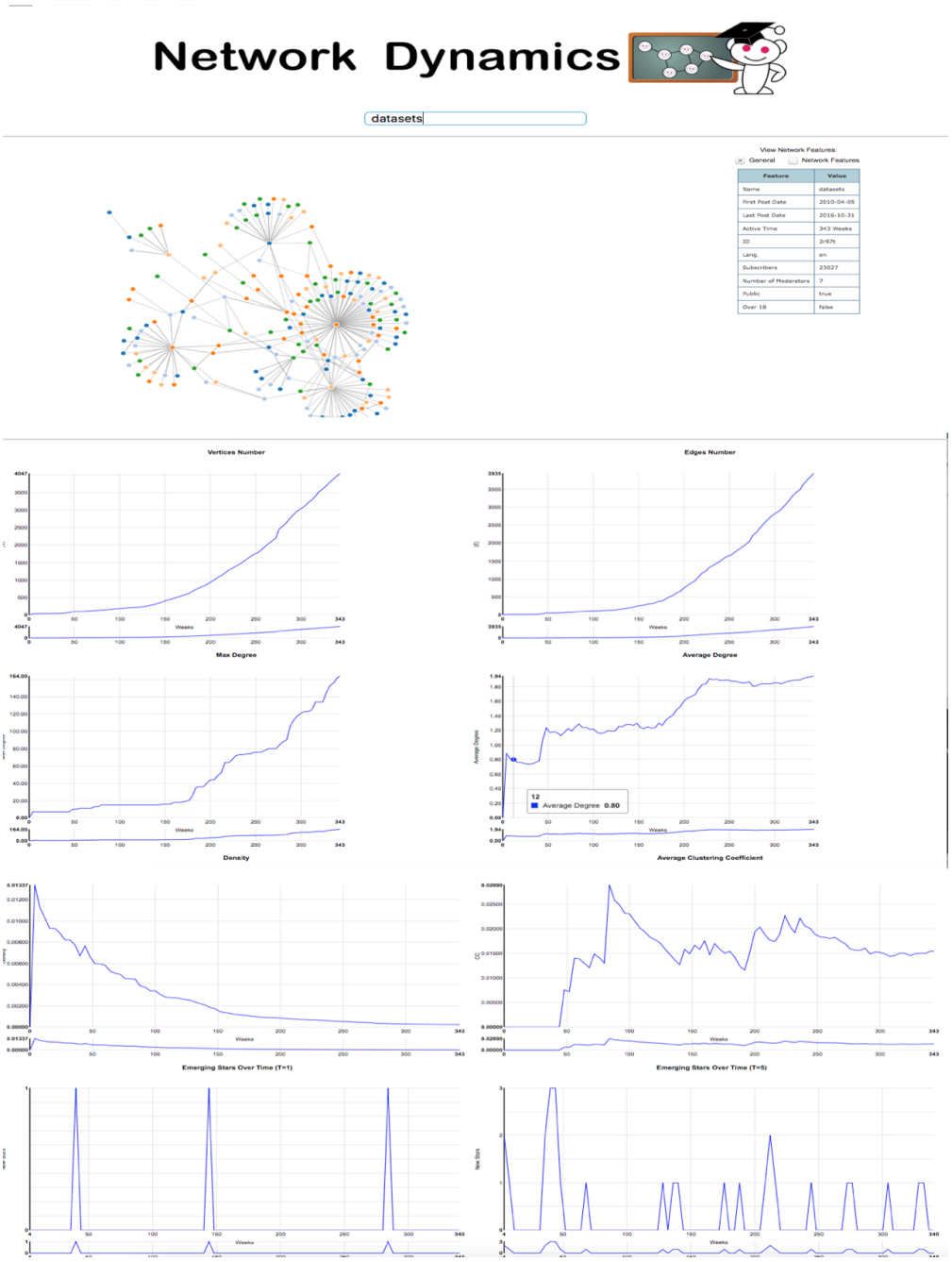

Figure 9. We have developed an interactive website that makes it possible to view and interact directly with the study's data.

[5] Steven H Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.

[6] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[7] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[8] Michael Fire, Lena Tenenboim-Chekina, Rami Puzis, Ofrit Lesser, Lior Rokach, and Yuval Elovici. Computationally efficient link prediction in a variety of social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):10, 2013.

[9] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[10] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pages 243–246. ACM, 2015.

[11] Reddit comments dataset.

[12] Fics games database. http://www.ficsgames.org/. Accessed: 2017-01-03.

[13] WikiTree. Wikitree database dumps. https://www.wikitree.com/wiki/Database_Dumps. Accessed: 2017-01-03.

[14] Ivan Brugere. Bitcoin transaction network dataset. http://compbio.cs.uic.edu/data/bitcoin/. Accessed: 2017-01-03.

[15] D Price. Statistical studies of networks of scientific papers. In *Statistical Association Methods for Mechanized Documentation: Symposium Proceedings*, volume 269, page 187. US Government Printing Office, 1965.

[16] Derek de Solla Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American society for Information science*, 27(5):292–306, 1976.

[17] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[18] Arnaud Sallaberry, Faraz Zaidi, and Guy Melançon. Model for generating artificial social networks having community structures with small-world and scale-free properties. *Social Network Analysis and Mining*, 3(3):597–609, 2013.

[19] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[20] Sergey N Dorogovtsev, José Fernando F Mendes, and Alexander N Samukhin. Structure of growing networks with preferential linking. *Physical review letters*, 85(21):4633, 2000.

[21] Petter Holme and Beom Jun Kim. Growing scale-free networks with tunable clustering. *Physical review E*, 65(2):026107, 2002.

[22] Mark Newman, Albert-Laszlo Barabasi, and Duncan J Watts. *The structure and dynamics of networks*. Princeton University Press, 2011.

[23] Sergei N Dorogovtsev and José FF Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. OUP Oxford, 2013.

[24] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2008.

[25] Kelly Bergstrom. "don't feed the troll": Shutting down debate about community expectations on reddit. com. *First Monday*, 16(8), 2011.

[26] Sanjay Ram Kairam, Dan J Wang, and Jure Leskovec. The life and death of online groups: Predicting group growth and longevity. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 673–682. ACM, 2012.

[27] Wikitree profile manager. https://www.wikitree.com/wiki/Profile_Manage. Accessed: 2017-06-15.

[28] Wikitree trusted list. https://www.wikitree.com/wiki/Trusted_List. Accessed: 2017-06-15.

[29] Michael Fire and Yuval Elovici. Data mining of online genealogy datasets for revealing lifespan patterns in human population. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(2):28, 2015.

[30] Kdd cup 2016. https://kddcup2016.azurewebsites.net/Data. Accessed: 2017-06-15.

[31] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer, 2013.

[32] Jari Saramäki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and Janos Kertesz. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105, 2007.

[33] DG Hyams. Curveexpert software. http://www.curveexpert.net, 2010.

[34] Jack Hessel, Chenhao Tan, and Lillian Lee. Science, askscience, and badscience: On the coexistence of highly related communities. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[35] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment*, 5(8):716–727, 2012.

[36] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.

[37] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *HotCloud*, 10(10-10):95, 2010.

[38] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

1. Supplementary Multimedia Links

A.1 External Datasets

The following project's datasets are available at the project's website:

Dataset S1. The Reddit networks' evolution dataset. This dataset contains the evolution over time of 20,128 subreddits and their corresponding 1,023,995 graphs (about 478 GB of compressed data).

Dataset S2. The Free Internet Chess Server network's evolution dataset (about 6.4 GB of compressed data).

Dataset S3. The co-authorship networks' evolution dataset. This dataset contains the co-authorship 9,005 networks of research fields and their corresponding 770,854 graphs (about 419 GB of compressed data).

Dataset S4. The citations networks' evolution dataset. This dataset contains the citation networks of 8,996 research fields and their corresponding 769,793 graphs (about 29 GB of compressed data).

Dataset S5. The Bitcoin Transaction network's evolution dataset (about 0.6 GB of compressed data).

Dataset S6. The Reddit networks' final graphs dataset. This dataset contains the final graph instance of 20,128 subreddits in October 2016 (about 25 GB of compressed data).

Dataset S7. The Join-Rate-Curves dataset. This dataset contains 38,129 times-series of the co-authorship, citation, and subreddit JRCs analyzed in this study.

A.2 Code Tutorials

The following code tutorial are available at the project's website:

Code Tutorial S1. Analyzing the social networks of over 2.7 billion Reddit comments. A Jupyter Notebook code tutorial explains and demonstrates how we analyzed the Reddit dataset.

Code Tutorial S2. The TPA model code. A Jupyter Notebook code tutorial provides explanations of how to create random complex networks using the TPA model.

A.3 Figure Collections

The following figure collections are available at the project's website:

Figure Collection S1. The citation networks' Join-Rate-Curves. This dataset consists of 8,996 citation network JRCs ordered by the networks' vibrancies in descending order.

Figure Collection S2. The co-authorship networks' Join-Rate-Curves. This dataset consists of 9,005 co-authorship network JRCs ordered by the networks' vibrancies in descending order.

Figure Collection S3. The subreddit networks' Join-Rate-Curves. This dataset consists of 20,128 subreddit network JRCs ordered by the networks' vibrancies in descending order.

A.4 Supplementary Video

Video S1. The Rise and Fall of Network Stars Video. This 4:10-minute video provides an overview of both the main research results and of the TPA model.