PNAS
www.pnas.org

Supplementary Information for

Simplicial closure and higher-order link prediction

Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, Jon Kleinberg

Jon Kleinberg.
E-mail: kleinber@cs.cornell.edu

This PDF file includes:

Supplementary text
Figs. S1 to S3
Tables S1 to S7
References for SI reference citations

Supporting Information Text

For ease of reading, we start each section in the supporting information on its own page.

Dataset collection and construction

Here we provide a more complete description of the datasets used in the main text.

Coauthorship. In these datasets, the nodes correspond to authors, and each simplex represents the authors on a scientific publication. The timestamp is the year of publication. We analyze three coauthorship networks—one derived from DBLP, an online bibliography for computer science, and two derived from the Microsoft Academic Graph (MAG). We used the September 3, 2017 release of DBLP[*]) and the MAG version released with the Open Academic Graph[†] (1). We constructed two field-specific datasets by filtering the MAG data according to keywords in the "field of study" information. One dataset consisted of all papers with "History" as a field of study and the other all papers with "Geology" as a field of study.

Stack Exchange tags. Stack Exchange is a collection of question-and-answer web sites.[‡] Users post questions and may annotate each question with up to 5 tags that specify topic areas spanned by the question. We derive tag networks where nodes correspond to tags and each simplex represents the tags on a question. The timestamp for a simplex is the time that the question was posted on the web site. We derived three datasets corresponding to three stack exchange web sites: Stack Overflow,[§] Mathematics Stack Exchange,[¶] and Ask Ubuntu.[‖] The raw data was downloaded from the Stack Exchange data dump,[**] which contains the complete history of the content on the stack exchange web sites.

Stack Exchange threads. We also formed user interaction datasets from the stack exchange web sites. Users post answers to questions, creating a question-and-answer "thread." The nodes are users and simplices correspond to the users asking a question or posting an answer on a single thread. We only considered threads where the question and all answers were posted within 24 hours. The timestamps of the simplices are the times that the question was posted.

National Drug Code Directory (NDC). Under the Drug Listing Act of 1972, the U.S. Food and Drug Administration releases information on all commercial drugs going through the regulation of the agency. We constructed two datasets from this data where simplices correspond to drugs. In one, the nodes are classification labels (e.g., serotonin reuptake inhibitor), and simplices are comprised of all labels applied to a drug; in the other, the nodes are substances (e.g., testosterone) and simplices are constructed from all substances in a drug. In both derived datasets, the timestamps are the days when the drugs were first marketed.

United States Congress. We derived two datasets from political networks, where the nodes are congresspersons in the U.S. congress. In the first, simplices represent all members of committees and sub-committees in the House of Representatives (Congresses 101 to 107, from 1989 to 2003), and the timestamp of the simplex is the year that the committee formed (2, 3). In the second dataset, simplices are comprised of the sponsor and co-sponsors of legislative bills put forth in the House of Representatives and the Senate (4, 5), and the timestamps are the days that the bills were introduced.

Email. In email communication, messages can be sent to multiple recipients. We analyze two email datasets—one from communication between Enron employees (6) and the other from a European research institution (7). In both datasets, nodes are email addresses. In the Enron dataset, a simplex consists of the sender and all recipients of an email. The data source for the European research institution only contains (sender, receiver, timestamp) tuples, where timestamps are recorded at 1-second resolution (7). Simplices consist of a sender and all receivers such that the email between the two has the same timestamp.

Human contact. The human contact networks are constructed from interactions recorded by wearable sensors in a high school (8) and a primary school (9). The sensors record proximity-based contacts every 20 seconds. We construct a graph for each interval, where nodes i and j are connected if they are in contact during the interval. Simplices are all maximal cliques in the graph at each time interval.

DAWN. The Drug Abuse Warning Network (DAWN) is a national health surveillance system that records drug use contributing to hospital emergency department visits throughout the United States. Simplices in our dataset are the drugs used by a patient (as reported by the patient) in an emergency department visit. The drugs include illicit substances, prescription and over-the-counter medication, and dietary supplements. Timestamps of visits are recorded at the resolution of quarter-years, spanning a total duration of 8 years. For a period of time, the recording system only recorded the first 16 drugs reported by a patient, so we only use (at most) the first 16 drugs reported by a patient for the entire dataset.

Music collaboration. Musical artists often collaborate on individual songs. We derive a dataset where nodes are artists and a simplex consists of all artists collaborating on a song. The songs were obtained from a web crawl of the music lyrics web site Genius.[††] We consider the collaborating artists to be the lead artist along with any "featured" artists (this excludes some cases where lyrics from an artist are included but that artist is not listed as a featured artist). The timestamps are the release dates of the songs. We only collected data from songs that contain the "rap" tag on the web site and discarded songs without a specified release date. The crawler ran for several days and collected over 500,000 songs.

[*] http://dblp.org/xml/release/

[†] https://www.openacademic.ai/oag/

[‡] https://stackexchange.com

[§] https://stackoverflow.com

[¶] https://math.stackexchange.com

[‖] https://askubuntu.com

[**] https://archive.org/details/stackexchange; downloaded September 20, 2017.

[††] https://genius.com/

Table S1. Temporal asynchrony and open triangles. For each open triangle in each dataset, we find the number of overlaps between the active intervals of the three edges, where an active interval of an edge has end points given by the earliest and latest timestamps of simplices containing the two nodes in the edge (Eq. (1)). The edges in most open triangles have three pairwise overlapping intervals. In these cases, there is a time period where all three edges were simultaneously active by Helly's theorem.

		# overlaps			
Dataset	# open triangles	0	1	2	3
coauth-DBLP	1,295,214	0.012	0.143	0.123	0.722
coauth-MAG-history	96,420	0.002	0.055	0.059	0.884
coauth-MAG-geology	2,494,960	0.010	0.128	0.109	0.753
tags-stack-overflow	300,646,440	0.002	0.067	0.071	0.860
tags-math-sx	2,666,353	0.001	0.040	0.049	0.910
tags-ask-ubuntu	3,288,058	0.002	0.088	0.085	0.825
threads-stack-overflow	99,027,304	0.001	0.034	0.037	0.929
threads-math-sx	11,294,665	0.001	0.038	0.039	0.922
threads-ask-ubuntu	136,374	0.000	0.020	0.023	0.957
NDC-substances	1,136,357	0.020	0.196	0.151	0.633
NDC-classes	9,064	0.022	0.191	0.136	0.652
DAWN	5,682,552	0.027	0.216	0.155	0.602
congress-committees	190,054	0.001	0.046	0.058	0.895
congress-bills	44,857,465	0.003	0.063	0.113	0.821
email-Enron	3,317	0.008	0.130	0.151	0.711
email-Eu	234,600	0.010	0.131	0.132	0.727
contact-high-school	31,850	0.000	0.015	0.019	0.966
contact-primary-school	98,621	0.000	0.012	0.014	0.974
music-rap-genius	70,057	0.028	0.221	0.141	0.611

Temporal asynchrony and open triangles

Our datasets contain temporal dynamics, so edges may only be "active" for certain periods in the total time spanned by the dataset. This provides one plausible explanation for the existence of open triangles. For example, in coauthorship networks, an open triangle may arise when three separate collaborations occurred in disjoint time periods. To investigate the importance of such effects, we analyze the temporal asynchrony in open triangles in our datasets. Let the "active interval" of an edge in the projected graph be the interval bounded by the earliest and latest timestamps of simplices containing the two nodes in the edge. Recall that our datasets are defined by a collection of timestamped simplices $\{(S_i, t_i)\}$, where each $S_i \subset V$ is the simplex and each $t_i \in \mathbb{R}$ is a timestamp. The active interval of an edge (u, v) is then

$$I_{u,v} = [\min\{t_i \mid u, v \in S_i\},\ \max\{t_i \mid u, v \in S_i\}]. \qquad [1]$$

For each open triangle in each dataset, we compute the number of pairwise overlapping active intervals amongst the three edges in the triangle (Table S1). In the majority of cases, all three pairs of intervals overlap. By Helly's theorem, there is an interval of time for which all three edges in the open triangle are simultaneously active. Stated differently, in the coauthorship example, the collaborators could have theoretically formed a closed triangle during this time period, but they did not. We conclude that temporal asynchrony is not a major reason for the presence of open triangles in our datasets.

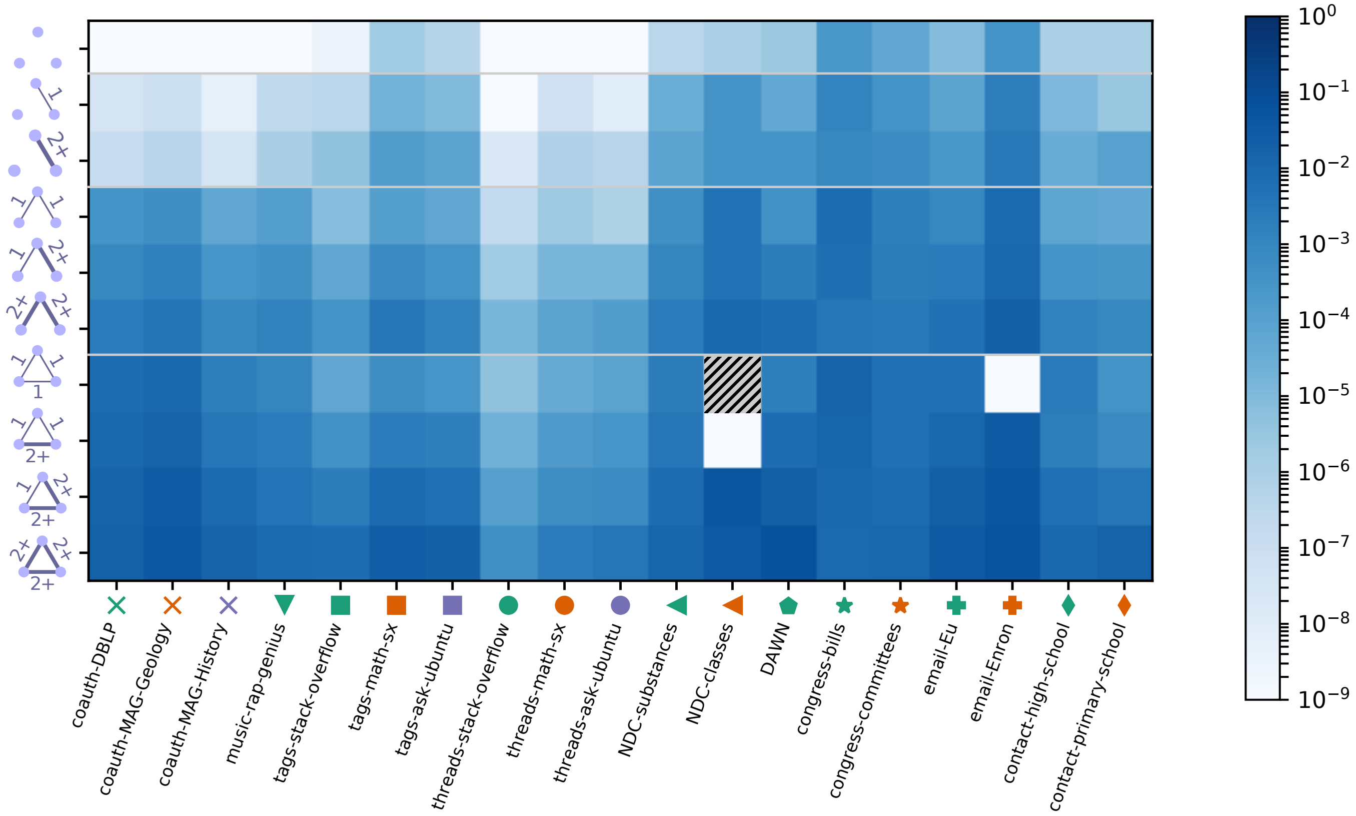


Fig. S1. Heat map of the probabilities of simplicial closure events as a function of the 3-node open configuration. We use the first 80% of the timestamped data to determine the configuration of every 3-node set and compute the probability that the set appears in a simplex in the final 20% of the data, conditioned on the open configuration. Shaded boxes are configurations that appear 20 or fewer times in the first 80% of the data. The four sections of the heat map correspond to 0, 1, 2, or 3 edges in the induced subgraph.

Simplicial closure probabilities

We now present simplicial closure event probabilities on three and four nodes. The setup is the same as the main text. We split the data into training and test sets, corresponding to the first 80% and final 20% of time-ordered simplices, respectively. Given the configuration of a set of three or four nodes in the training data, we measure the probability that the nodes go through a simplicial closure event in the final 20% of the data.

In the case of 3-node simplicial closure events, we determine the open configuration in the training set by examining the three nodes in the projected graph. Effectively, we examined how many times each of the three 2-*node subsets* co-appeared in a simplex. Figure S1 shows a heat map of the closure probabilities as a function of the open 3-node configuration.

For 4-node open configurations we proceed analogously, using the corresponding 3-*node subsets*. Specifically, for a given set of 4 nodes, every triangle in the projected graph is classified as either (i) an open simplicial tie, i.e., the triangle is open; (ii) a weak simplicial tie, meaning that the three nodes have appeared in just one simplex together; or (iii) a strong simplicial tie, meaning that the three nodes have appeared in at least two simplices together. In contrast to the 3-node case, these 4-node configurations are not completely determined by the weighted projected graph, since the projected graph (as defined) does not contain information on whether or not 3 nodes induce a closed triangle. Thus, with 4-node simplices, we make use of additional topological information provided in our set-valued datasets.

One could also account for the tie strengths of the 2-node subsets (edges) in a 4-node configuration—a complete characterization of the possible induced configurations leading to a simplicial closure event is much more complex for the 4-node case than the 3-node case. For an accessible study on the closure patterns of 4-node simplices, we measure the probability of closure with respect to the 27 open configurations where the induced 4-node subgraph in the projected graph contains at least one triangle. Our classification distinguishes triangles by tie strength (open, weak, or closed), but not by edge tie strengths, other than by what is implied by the triangles. Figure S2 shows a heat map of the closure probabilities as a function of the open 4-node configuration.

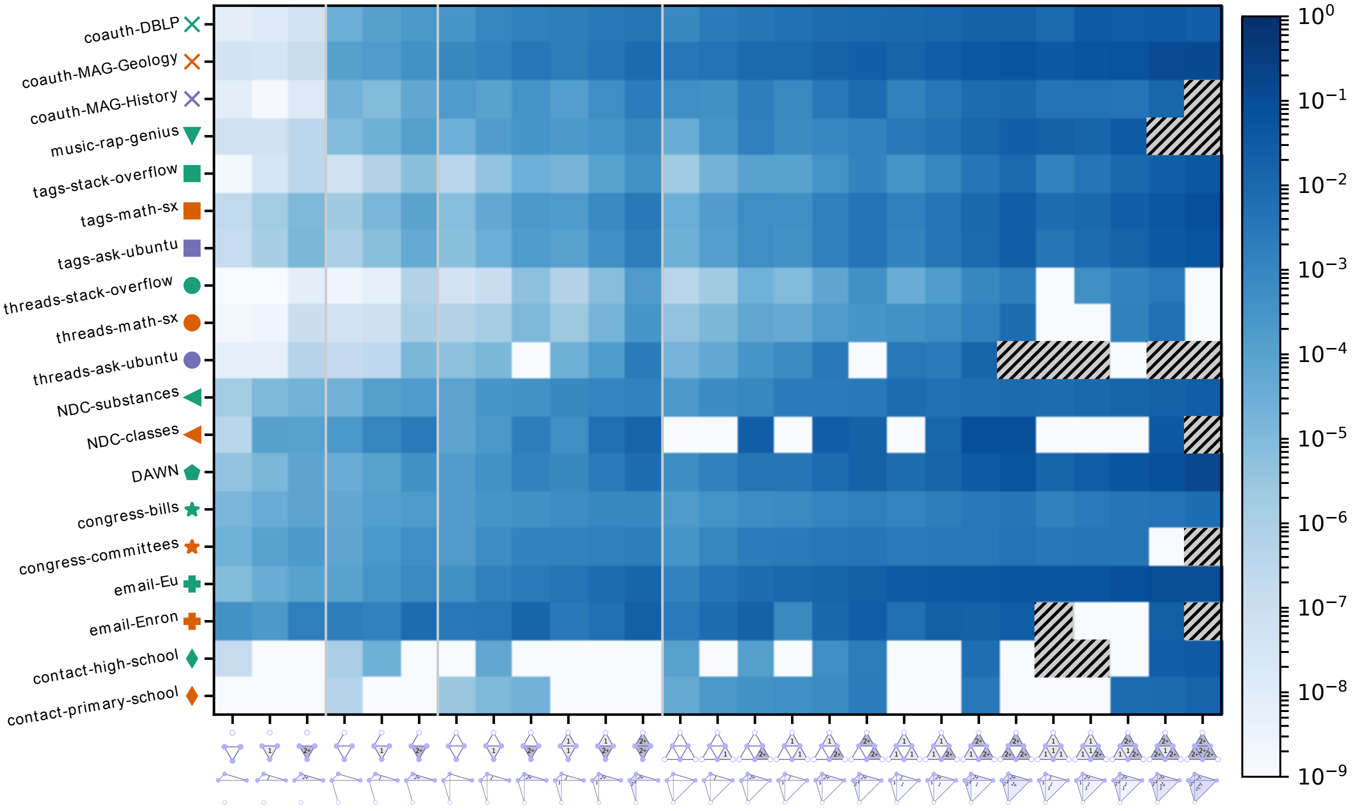


Fig. S2. Heat map of the probabilities of simplicial closure events as a function of the 4-node open configuration. We use the first 80% of the timestamped data to determine the configuration of every 4-node set that contains at least 1 triangle and does not appear in a simplex. We then compute the probability that a 4-node set appears in a simplex in the final 20% of the data, conditioned on the open configuration. In an open configuration, there are three types of simplicial tie strengths for a triangle—open, weak, and strong—given by the number of times the three nodes in the triangle have co-appeared in a simplex (zero, one, or at least two times). Shaded boxes are configurations that appear 20 or fewer times in the first 80% of the data. We illustrate each subgraph configuration on the x-axis with a projection of the simplex onto two dimensions (top line—the unfilled circle represents the same node) as well as a tetrahedral three-dimensional perspective figure (bottom line). The four sections of the heat map correspond to 3, 4, 5, or 6 edges in the configuration.

Table S2. Consistency in the effects of tie strength and edge density in 3-node configurations on simplicial closure events at different points in time. For edge density, we tested whether or not the closure event probability of a fixed weighted induced subgraph configuration and the same configuration with an additional unit-weight edge significantly increases or decreases the closure probability (at significance level 10^{-5}). For tie strength, we tested whether the closure event probability significantly increases or decreases when comparing a fixed weighted induced subgraph containing at least one weak tie, and the same configuration where the weak tie is converted to a strong tie (edge weight at least 2 in the projected graph). The "total" column is the number of tested hypotheses. We apply the tests to filtered datasets that only contain the first X% of timestamped simplices (in time order). We only consider cases where the configuration has at least 25 samples in the first 80% of timestamped simplices of a filtered dataset. Increasing either edge density or tie strength significantly increases the closure probability for all values of X, suggesting that these features are positive indicators of simplicial closure over time.

	edge density increases			tie strength increases		
	sig. incr.	sig. decr.	total	sig. incr.	sig. decr.	total
X						
40	89	0	113	71	2	113
60	101	0	113	80	7	113
80	102	0	113	86	2	113
100	96	0	107	76	6	107

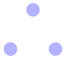
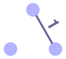
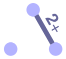
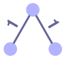
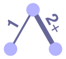
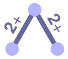
Simplicial closure events at different points in time

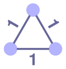
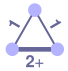
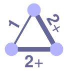
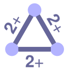
In the main text, we studied simplicial closure events by counting the 3-node and 4-node configuration patterns in the first 80% of timestamped simplices and then measuring the fraction of instances that experience a simplicial close event in the final 20% of timestamped simplices. Here, we show that our results are consistent when examining different time slices of the data. We first filtered each dataset to contain only the first X% of timestamped simplices, for $X = 40, 60, 80$ (the original dataset is the case of $X = 100$.) We then split the filtered dataset into the first 80% and last 20% of timestamped simplices (within the time frame of the filtered dataset) and computed the probabilities of simplicial closure events.

Table S3 lists the 3-node simplicial closure event probabilities as a function of the configuration of the 3 nodes in the first 80% of the data for $X = 40, 60, 80, 100$, and Fig. S3 provides the heat maps for each value of X (analogous to the heat map in Fig. S1). Broadly, the closure probabilities remain similar for different values of X. We also find that edge density and tie strength are always positive indicators of simplicial closure events, regardless of X (Table S2). Thus, these features are important for simplicial closure events throughout the history of the network dynamics.

The tension between these features is also consistent over time. The weak open triangle (where all three edges are weak ties) is more likely to close than the strong wedge (the 3-node configuration with exactly two strong ties) in the coauth-DBLP, coauth-MAG-Geology, and congress-bills datasets for all values of X as well as in the congress-committees dataset for $X = 60, 80, 100$. On the other hand, the strong wedge is more likely to close in the three stack exchange tags networks, DAWN, and threads-stack-overflow for all values of X as well as in threads-math-sx for $X = 80, 100$.

Table S3. Simplicial closure event probabilities of different configurations at different points in time. We first filtered each dataset to contain only the first X% of timestamped simplices, for $X = 40, 60, 80, 100$. We then split the filtered dataset into the first 80% and last 20% of timestamped simplices (within the time frame of the filtered dataset). We record the probability of closure in last 20% conditioned on the open configuration in the first 80%.

																								
	40	60	80	100	40	60	80	100	40	60	80	100	40	60	80	100	40	60	80	100	40	60	80	100
coauth-DBLP	8.2e-13	1.2e-12	8.3e-13	9.3e-13	3.1e-08	4.2e-08	3.4e-08	3.6e-08	1.1e-07	1.5e-07	1.2e-07	1.3e-07	4.4e-04	5.2e-04	3.8e-04	3.5e-04	9.7e-04	1.2e-03	9.4e-04	8.8e-04	2.0e-03	2.5e-03	2.2e-03	2.1e-03
coauth-MAG-Geology	7.9e-12	5.0e-12	3.2e-12	4.2e-12	1.1e-07	9.9e-08	7.6e-08	8.9e-08	4.1e-07	4.6e-07	3.6e-07	4.5e-07	5.8e-04	6.8e-04	5.1e-04	5.3e-04	1.4e-03	1.8e-03	1.5e-03	1.6e-03	3.0e-03	4.4e-03	3.7e-03	4.1e-03
coauth-MAG-History	1.2e-12	8.8e-13	3.3e-13	1.8e-13	2.7e-08	2.0e-08	1.0e-08	5.6e-09	1.6e-07	1.4e-07	6.3e-08	3.9e-08	1.4e-04	1.8e-04	1.0e-04	6.3e-05	6.2e-04	8.5e-04	4.1e-04	2.9e-04	1.3e-03	2.3e-03	2.5e-03	1.0e-03
music-rap-genius	1.6e-09	8.6e-10	4.3e-10	1.2e-10	1.3e-06	6.2e-07	5.7e-07	2.3e-07	5.5e-06	2.9e-06	1.9e-06	1.1e-06	3.3e-04	2.2e-04	2.2e-04	1.3e-04	1.0e-03	8.0e-04	5.6e-04	4.4e-04	3.3e-03	2.9e-03	1.7e-03	1.3e-03
tags-stack-overflow	3.6e-09	3.5e-09	2.9e-09	2.8e-09	4.8e-07	4.5e-07	3.8e-07	3.8e-07	5.0e-06	4.6e-06	4.2e-06	4.2e-06	9.6e-06	8.5e-06	7.4e-06	7.4e-06	6.9e-05	6.3e-05	5.8e-05	5.7e-05	4.6e-04	4.1e-04	3.8e-04	3.7e-04
tags-math-sx	1.0e-06	1.4e-06	1.9e-06	1.9e-06	1.7e-05	1.7e-05	1.9e-05	2.1e-05	1.1e-04	1.1e-04	1.5e-04	1.4e-04	1.3e-04	1.2e-04	1.3e-04	1.2e-04	7.0e-04	7.0e-04	7.2e-04	6.9e-04	3.4e-03	3.3e-03	3.4e-03	3.2e-03
tags-ask-ubuntu	4.9e-07	5.4e-07	7.9e-07	4.8e-07	1.1e-05	1.3e-05	1.4e-05	9.8e-06	8.1e-05	9.7e-05	1.1e-04	7.9e-05	9.4e-05	8.8e-05	8.6e-05	6.2e-05	4.8e-04	4.6e-04	4.7e-04	3.8e-04	1.5e-03	1.6e-03	1.5e-03	1.4e-03
threads-stack-overflow	3.2e-12	8.4e-13	4.3e-13	2.2e-13	5.5e-09	2.2e-09	1.4e-09	9.0e-10	8.4e-08	4.2e-08	2.7e-08	2.1e-08	6.4e-07	3.3e-07	2.5e-07	1.9e-07	4.8e-06	3.0e-06	2.3e-06	2.0e-06	2.5e-05	1.7e-05	1.5e-05	1.5e-05
threads-math-sx	2.3e-10	1.4e-10	6.4e-11	4.2e-11	1.5e-07	1.3e-07	7.8e-08	6.0e-08	1.2e-06	1.3e-06	9.4e-07	7.4e-07	3.6e-06	3.8e-06	2.5e-06	2.3e-06	1.6e-05	2.0e-05	1.7e-05	1.5e-05	6.4e-05	8.9e-05	9.0e-05	7.9e-05
threads-ask-ubuntu	5.3e-11	1.4e-11	5.3e-12	3.2e-12	2.8e-08	2.1e-08	1.2e-08	9.0e-09	5.7e-07	5.2e-07	5.7e-07	4.1e-07	1.7e-06	5.6e-07	7.7e-07	7.8e-07	1.6e-05	1.4e-05	1.8e-05	1.6e-05	6.8e-05	1.0e-04	1.6e-04	1.4e-04
NDC-substances	1.4e-06	3.4e-06	9.6e-07	3.8e-07	1.1e-04	1.9e-04	7.5e-05	3.3e-05	1.9e-04	4.2e-04	1.9e-04	7.8e-05	2.5e-03	2.9e-03	1.4e-03	4.8e-04	6.0e-03	6.0e-03	3.2e-03	1.1e-03	1.0e-02	1.2e-02	6.7e-03	2.2e-03
NDC-classes	5.3e-07	9.0e-06	1.5e-06	8.4e-07	3.7e-06	8.8e-04	1.7e-04	3.1e-04	3.6e-04	1.8e-03	7.7e-04	3.4e-04	0.0e+00	4.0e-02	0.0e+00	4.8e-03	0.0e+00	7.1e-02	3.0e-03	4.8e-03	9.1e-03	5.4e-02	2.4e-02	1.1e-02
DAWN	1.5e-06	2.1e-06	2.7e-06	2.5e-06	4.2e-05	4.6e-05	6.0e-05	5.4e-05	2.1e-04	2.7e-04	3.5e-04	3.4e-04	3.4e-04	3.9e-04	4.7e-04	4.1e-04	1.6e-03	1.8e-03	2.2e-03	2.1e-03	5.2e-03	6.3e-03	7.6e-03	7.3e-03
congress-bills	1.9e-04	9.2e-04	3.0e-04	2.5e-04	7.6e-04	3.0e-03	1.2e-03	1.3e-03	9.9e-04	2.4e-03	1.2e-03	9.1e-04	4.2e-03	1.2e-02	5.4e-03	8.1e-03	5.4e-03	1.0e-02	5.2e-03	6.1e-03	5.0e-03	7.2e-03	3.4e-03	3.8e-03
congress-committees	0.0e+00	1.5e-04	3.7e-05	5.8e-05	0.0e+00	6.7e-04	2.9e-04	3.6e-04	0.0e+00	9.9e-04	5.7e-04	6.3e-04	0.0e+00	2.4e-03	1.5e-03	1.7e-03	0.0e+00	3.2e-03	2.2e-03	2.1e-03	0.0e+00	3.4e-03	3.0e-03	3.1e-03
email-Eu	8.2e-06	1.5e-05	1.4e-05	8.4e-06	1.3e-04	1.8e-04	1.4e-04	8.3e-05	3.3e-04	3.6e-04	5.0e-04	2.4e-04	1.7e-03	1.1e-03	1.2e-03	1.0e-03	3.6e-03	3.3e-03	3.9e-03	2.4e-03	7.8e-03	6.4e-03	8.1e-03	5.2e-03
email-Enron	6.3e-04	4.3e-04	3.8e-04	3.4e-04	5.4e-03	2.2e-03	1.8e-03	1.9e-03	4.1e-03	4.3e-03	3.3e-03	3.1e-03	1.9e-02	1.1e-02	1.5e-02	9.4e-03	2.4e-02	2.6e-02	2.3e-02	1.2e-02	2.4e-02	3.9e-02	2.5e-02	2.1e-02
contact-high-school	6.4e-07	1.1e-06	1.1e-06	9.4e-07	1.5e-05	1.6e-05	7.5e-06	1.2e-05	5.3e-05	4.3e-05	8.6e-05	3.7e-05	3.8e-04	0.0e+00	9.1e-05	7.2e-05	1.1e-03	4.1e-04	6.7e-04	3.5e-04	2.4e-03	1.6e-03	2.1e-03	1.4e-03
contact-primary-school	2.6e-06	0.0e+00	3.2e-05	1.0e-06	3.7e-06	1.7e-05	6.9e-05	3.2e-06	6.7e-05	5.6e-05	3.2e-04	1.0e-04	9.0e-05	0.0e+00	1.9e-04	5.1e-05	2.7e-04	2.6e-04	8.6e-04	2.6e-04	1.4e-03	9.9e-04	2.1e-03	8.8e-04

																
	40	60	80	100	40	60	80	100	40	60	80	100	40	60	80	100
coauth-DBLP	8.3e-03	8.6e-03	8.5e-03	7.6e-03	1.0e-02	1.1e-02	1.1e-02	1.1e-02	1.2e-02	1.5e-02	1.5e-02	1.7e-02	1.3e-02	1.6e-02	1.7e-02	1.9e-02
coauth-MAG-Geology	8.6e-03	1.2e-02	9.4e-03	1.0e-02	1.4e-02	2.0e-02	1.5e-02	1.7e-02	1.7e-02	2.9e-02	2.3e-02	2.7e-02	2.2e-02	3.7e-02	3.0e-02	4.0e-02
coauth-MAG-History	1.7e-03	3.4e-03	1.6e-03	1.9e-03	6.0e-03	9.1e-03	4.9e-03	3.8e-03	8.3e-03	2.1e-02	1.4e-02	9.1e-03	5.1e-03	3.6e-02	3.8e-02	1.5e-02
music-rap-genius	1.4e-03	2.3e-03	1.3e-03	1.1e-03	4.8e-03	2.9e-03	2.8e-03	2.2e-03	1.2e-02	7.9e-03	6.5e-03	4.6e-03	2.0e-02	1.7e-02	1.2e-02	8.6e-03
tags-stack-overflow	9.2e-05	7.0e-05	6.5e-05	6.5e-05	5.2e-04	4.4e-04	4.0e-04	3.9e-04	2.7e-03	2.3e-03	2.1e-03	2.1e-03	9.6e-03	8.4e-03	7.7e-03	7.6e-03
tags-math-sx	6.3e-04	5.6e-04	5.4e-04	5.6e-04	2.4e-03	2.3e-03	2.3e-03	2.1e-03	1.0e-02	9.1e-03	9.2e-03	8.6e-03	2.9e-02	2.7e-02	2.7e-02	2.6e-02
tags-ask-ubuntu	5.3e-04	4.5e-04	3.3e-04	2.9e-04	2.2e-03	2.2e-03	1.8e-03	1.7e-03	6.9e-03	7.1e-03	5.9e-03	5.9e-03	2.1e-02	2.3e-02	1.8e-02	1.9e-02
threads-stack-overflow	9.6e-06	6.1e-06	5.7e-06	4.9e-06	4.2e-05	3.5e-05	2.6e-05	2.5e-05	1.5e-04	1.3e-04	1.1e-04	1.1e-04	6.9e-04	5.8e-04	4.3e-04	4.7e-04
threads-math-sx	6.3e-05	5.9e-05	4.3e-05	4.0e-05	1.8e-04	2.3e-04	2.0e-04	1.7e-04	4.6e-04	6.4e-04	6.1e-04	5.4e-04	1.3e-03	2.3e-03	2.7e-03	2.2e-03
threads-ask-ubuntu	0.0e+00	0.0e+00	0.0e+00	8.4e-05	4.5e-05	9.2e-05	3.8e-04	2.9e-04	0.0e+00	5.2e-04	1.7e-03	6.5e-04	1.4e-03	2.2e-03	5.7e-03	3.6e-03
NDC-substances	1.5e-02	1.1e-02	7.6e-03	2.1e-03	3.1e-02	2.2e-02	1.3e-02	3.8e-03	4.8e-02	3.5e-02	2.2e-02	7.2e-03	7.3e-02	4.8e-02	3.9e-02	1.5e-02
NDC-classes	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	1.1e-01	0.0e+00	0.0e+00	1.3e-01	9.1e-02	5.2e-02	3.6e-02	0.0e+00	8.7e-02	3.4e-02
DAWN	1.7e-03	1.5e-03	2.3e-03	1.7e-03	5.7e-03	6.0e-03	7.9e-03	7.2e-03	1.5e-02	1.7e-02	2.2e-02	2.1e-02	4.8e-02	5.5e-02	7.3e-02	6.9e-02
congress-bills	7.9e-03	1.9e-02	9.5e-03	1.7e-02	9.5e-03	1.8e-02	1.1e-02	1.4e-02	8.4e-03	1.6e-02	1.0e-02	1.1e-02	9.6e-03	1.3e-02	1.1e-02	9.3e-03
congress-committees	0.0e+00	9.8e-03	6.1e-03	5.5e-03	0.0e+00	1.2e-02	8.5e-03	5.9e-03	0.0e+00	1.4e-02	1.1e-02	8.4e-03	0.0e+00	1.2e-02	1.6e-02	1.2e-02
email-Eu	2.5e-03	5.2e-03	1.2e-02	5.3e-03	1.6e-02	9.9e-03	1.5e-02	1.2e-02	2.6e-02	1.8e-02	2.4e-02	2.1e-02	4.7e-02	3.4e-02	4.8e-02	3.3e-02
email-Enron	0.0e+00	2.3e-02	0.0e+00	0.0e+00	6.6e-02	3.9e-02	7.6e-02	3.1e-02	5.9e-02	9.9e-02	8.2e-02	4.8e-02	9.2e-02	8.0e-02	1.4e-01	5.5e-02
contact-high-school	0.0e+00	0.0e+00	0.0e+00	2.6e-03	5.2e-03	2.2e-03	4.0e-03	1.8e-03	7.9e-03	4.9e-03	7.9e-03	5.7e-03	1.3e-02	1.2e-02	1.5e-02	1.2e-02
contact-primary-school	0.0e+00	0.0e+00	5.6e-04	3.8e-04	1.4e-03	1.3e-03	1.3e-03	7.9e-04	3.4e-03	3.6e-03	3.2e-03	3.4e-03	1.7e-02	1.8e-02	1.7e-02	1.7e-02

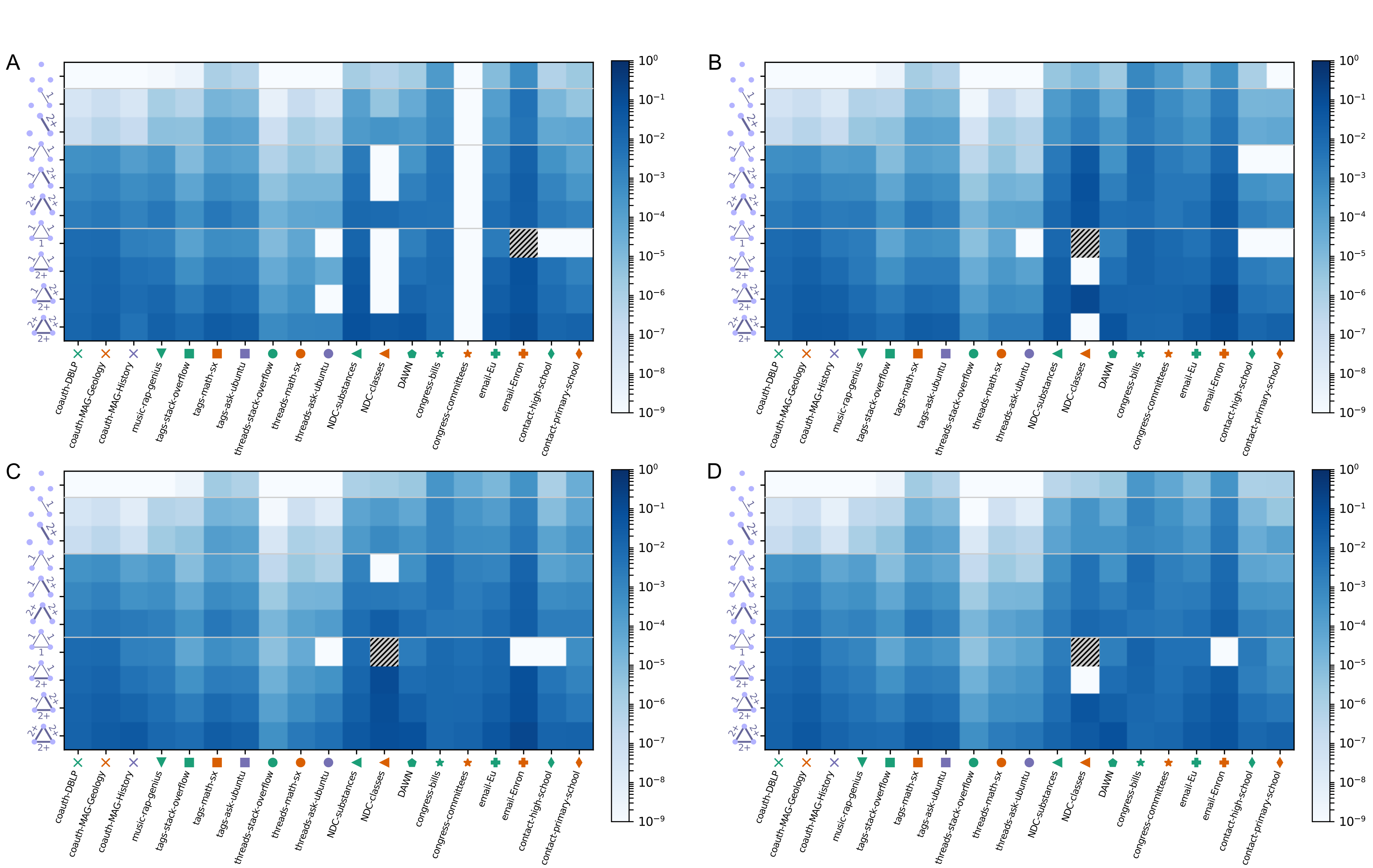
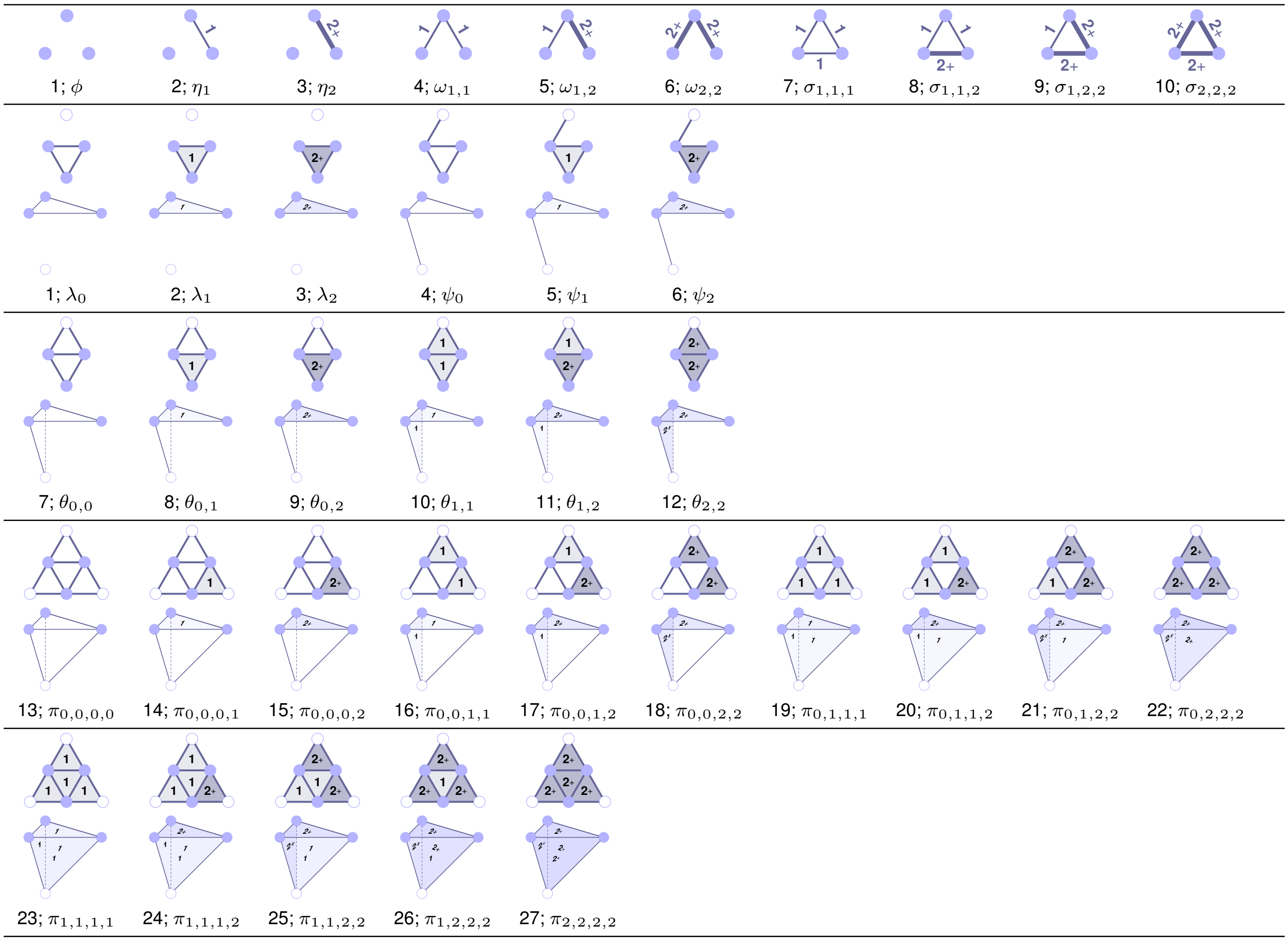


Fig. S3. Heat maps of simplicial closure event probabilities of different configurations at different points in time. We first filtered each dataset to contain only the first X% of timestamped simplices, for $X = 40$ (**A**), $X = 60$ (**B**), $X = 80$ (**C**), and $X = 100$ (**D**). We then split the filtered dataset into the first 80% and last 20% of timestamped simplices (within the time frame of the filtered dataset) and compute the probability of closure in last 20% conditioned on the open configuration in the first 80%. Overall, the four heat maps of simplicial closure event probabilities exhibit similar trends (actual probabilities are listed in Table S3). Shaded boxes are cases with fewer than 25 samples. The four sections of the heat map correspond to 0, 1, 2, or 3 edges in the induced subgraph.

Table S4. The 10 open 3-node configurations analyzed and the 27 open 4-node configurations analyzed in Figs. S1 and S2. We illustrate each 4-node configuration with a projection onto two dimensions (top—the unfilled circle represents the same node) as well as a tetrahedral three-dimensional perspective figure (bottom). For each configuration, we list (i) a reference number of the configuration and (ii) our notation for the number of instances of the configuration. For 3-node configurations, the subscripts 1 and 2 denote weak and strong simplicial ties, and for 4-node configurations, the subscripts 0, 1, and 2 denote open, weak, and strong simplicial ties. We also use $\tau_{i,j,k}$ to denote the sum of counts of open and closed 3-node, triangles ($1 \leq i \leq j \leq k \leq 2$) and $\rho_{i,j,k,l}$ to denote the sum of counts of open and closed 4-node, 6-edge tetrahedral wireframe configurations ($0 \leq i \leq j \leq k \leq l \leq 2$).



Efficient counting of simplicial closure event probabilities

An open configuration on three or four nodes is a set of nodes that have not jointly appeared in a simplex in the training set comprising the first 80% of the timestamped simplices (the training set). An instance of a subgraph configuration "closes" if the nodes subsequently all appear in one of the final 20% of timestamped simplices (the test set). For all newly formed simplices in the test set, we can check their prior configuration c in the training set, which provides the number of times each configuration closes. Dividing the number of closures of a configuration c by the total number of instances it was open in the training set gives the probability of a simplicial closure event. Most of the datasets we study are large enough that naively computing the simplicial closure event probabilities is infeasible. We need to develop efficient algorithms for computing the closure probabilities.

The key idea of our approach is that we do not need to *enumerate* all of the configurations in the training set and check if they close. Instead, we only need the *total count* of open configurations in the training data. We then count how many close by examining the test data directly. The idea of avoiding enumeration when simply counting suffices has been used in other fast graph configuration counting algorithms (7, 10).

Counting for 3-node configurations. We first show how to count the number of each 3-node subgraph configuration (the top row of Table S4). Recall that a weak tie corresponds to an edge in the projected graph with a weight of 1, whereas a strong tie corresponds to an edge with a weight of at least 2. Subscripts of 1 and 2 denote weak and strong ties in our notation. (Note that we use "2+" for strong ties in the illustrations in Table S4; however, it will be convenient to use the integer 2 in our description of the algorithms.)

Let $\tau_{i,j,k}$, $1 \le i \le j \le k \le 2$, be the number of (open or closed) triangles whose edges have the tie strengths given by the subscripts. For instance, $\tau_{1,1,1}$ is the number of triangles whose edges are all weak ties. Similarly, let $\sigma_{i,j,k}$ be the number of triangles with given tie strengths that are open (see the right-most configurations in the first row of Table S4). We can count the number of all triangles $\tau_{i,j,k}$ using a number of efficient triangle enumeration algorithms for sparse graphs (11). For each of these triangles we then determine whether it is closed by examining the entries of a simplex-to-node adjacency matrix (this can be efficiently read from our set-based data). The difference between the total number of triangles and the number of closed triangles gives us the open triangle counts $\sigma_{i,j,k}$.

Next, consider the number of 2-edge, 3-node induced "wedge" subgraphs. Let the symbols $\omega_{1,1}$, $\omega_{1,2}$, and $\omega_{2,2}$ denote these configurations, where the tie strengths of the two edges are given by the subscripts (see the first row of Table S4). Furthermore, let $d_1(u)$ and $d_2(u)$ be the number of weak and strong ties containing node u as an endpoint. Then $\omega_{i,j}$ is given by the number of (non-induced) 2-edge, 3-node subgraphs with tie strengths i and j minus the ones that appear in triangles:

$$\omega_{1,1} = \textstyle\sum_u \binom{d_1(u)}{2} - 3\tau_{1,1,1} - \tau_{1,1,2} \quad [2]$$

$$\omega_{2,2} = \textstyle\sum_u \binom{d_2(u)}{2} - 3\tau_{2,2,2} - \tau_{1,2,2} \quad [3]$$

$$\omega_{1,2} = \textstyle\sum_u d_1(u)d_2(u) - 2\tau_{1,1,2} - 2\tau_{1,2,2} \quad [4]$$

Now let η_1 and η_2 be the counts of the 1-edge, 3-node induced subgraphs, where again the tie strength of the edge is given by the subscript (see the first row of Table S4). Denote the total number of weak and strong ties by $m_s = \frac{1}{2}\sum_u d_s(u)$, $s = 1, 2$, and the total number of nodes by n. Then the total number of (non-induced) 1-edge, 3-node subgraphs with tie strength s is then $m_s(n-2)$. Induced 1-edge, 3-node subgraph are given by the non-induced counts minus the 2- and 3-edge induced counts discussed above:

$$\eta_1 = m_1(n-2) - 2\omega_{1,1} - \omega_{1,2} - 3\tau_{1,1,1} - 2\tau_{1,1,2} - \tau_{1,2,2} \quad [5]$$

$$\eta_2 = m_2(n-2) - 2\omega_{2,2} - \omega_{1,2} - 3\tau_{2,2,2} - 2\tau_{1,2,2} - \tau_{1,1,2} \quad [6]$$

Finally, let ϕ be the number of empty 3-node induced subgraphs of the projected graph (the top left of Table S4). The number of subsets of 3 nodes minus all other induced 3-node subgraphs gives the value of ϕ:

$$\phi = \binom{n}{3} - \sum_{s=1}^{2} \eta_s - \sum_{1 \le i,j \le 2} \omega_{i,j} - \sum_{1 \le i \le j \le k \le 2} \tau_{i,j,k}. \quad [7]$$

Counting for 4-node configurations. Now we describe how we compute the simplicial closure event probabilities conditioned on the 27 subgraph configurations on four nodes in Fig. S2 (these are the 4-node configurations in the second through fifth rows of Table S4). Recall that the simplicial tie strength of a triangle is (i) *open* if the three nodes form an open triangle; (ii) *weak* if the three nodes have jointly appeared in exactly one simplex; or (iii) *strong* if the three nodes have jointly appeared in at least two simplices. We use subscripts 0, 1, and 2 to denote these bins.

There are 15 total 4-node, 6-edge tetrahedral subgraph configurations. Each configuration corresponds to a non-decreasing 4-tuple of the simplicial tie strengths of the four triangles in the configuration. We denote the sum of open and closed tetrahedral counts by $\rho_{i,j,k,l}$, where i, j, k, and l denote the simplicial tie strengths, and the open tetrahedral counts by $\pi_{i,j,k,l}$ ($0 \le i \le j \le k \le l \le 2$; the 15 configurations in the bottom two row blocks of Table S4). We may count both $\rho_{i,j,k,l}$ and $\pi_{i,j,k,l}$ by enumerating 4-cliques using, e.g., the Chiba and Nishizeki algorithm (12) and checking if each 4-clique is closed or open by examining the simplex-node adjacency matrix.

Next, we consider counts of the six 4-node, 5-edge subgraph configurations $\theta_{i,j}$, where each configuration is given by a non-decreasing pair of simplicial tie strengths for the two triangles in the configuration (the third row block of Table S4). Each instance of this configuration consists of two triangles sharing one edge. We first use a fast triangle enumeration algorithm to compute matrices $Y^{(s)}$, $s \in \{0, 1, 2\}$, where $Y^{(s)}_{uv}$ is the number of triangles with simplicial tie strength s containing nodes u and v. The counts of the non-induced configuration, which we denote by $\theta'_{i,j}$, are then given by:

$$\theta'_{s,s} = \textstyle\sum_{(u,v)} \binom{Y^{(s)}_{uv}}{2}, \quad s = 0, 1, 2 \quad [8]$$

$$\theta'_{i,j} = \textstyle\sum_{(u,v)} Y^{(i)}_{uv} Y^{(j)}_{uv}, \quad 0 \le i < j \le 2. \quad [9]$$

The summations are over the edges (u, v) in the projected graph. Each non-induced instance of these subgraph configurations may correspond to a 6-edge tetrahedral configuration, and we need to adjust for these cases. Each (open or closed) 6-edge tetrahedron count $\rho_{i,j,k,l}$ contributes to the non-induced counts $\theta'_{i,j}$, $\theta'_{i,k}$, $\theta'_{i,l}$, $\theta'_{j,k}$, $\theta'_{j,l}$, and $\theta'_{k,l}$. To get the count $\theta_{i,j}$, we subtract the portion of $\theta'_{i,j}$ coming from the tetrahedra. Denote the set of valid 4-tuples of indices for the counts $\rho_{i,j,k,l}$ by $\mathcal{S}$. Formally, $\mathcal{S} = \{(i,j,k,l) \mid 0 \le i \le j \le k \le l \le 2\}$. Then $\theta_{i,j}$ is given by

$$\begin{aligned}\theta_{i,j} = \theta'_{i,j} &- \sum_{k,l \,:\, (i,j,k,l) \in \mathcal{S}} \rho_{i,j,k,l} - \sum_{k,l \,:\, (i,k,j,l) \in \mathcal{S}} \rho_{i,k,j,l} \\ &- \sum_{k,l \,:\, (i,k,l,j) \in \mathcal{S}} \rho_{i,k,l,j} - \sum_{k,l \,:\, (k,i,j,l) \in \mathcal{S}} \rho_{k,i,j,l} - \sum_{k,l \,:\, (k,i,l,j) \in \mathcal{S}} \rho_{k,i,l,j} - \sum_{k,l \,:\, (k,l,i,j) \in \mathcal{S}} \rho_{k,l,i,j}.\end{aligned} \quad [10]$$

Next, we show how to count 4-node, 4-edge subgraph configurations that contain one triangle. There are three such configurations, corresponding to the three possible simplicial ties in the triangle, and we denote the counts by ψ_s, $s \in \{0, 1, 2\}$ (the three right-most configurations in the second row of Table S4). We again compute non-induced counts and then subtract the induced counts of subgraphs with more edges, for which we showed how to compute above. Some additional notation will be helpful for these counts. Let $\mathcal{T}_s$ be the set of triangles with simplicial tie strength $s \in \{0, 1, 2\}$, and let a_s and b_s count how many times triangles with a particular tie strength appear in 5-edge configuration patterns and 6-edge configuration patterns:

$$\begin{aligned} a_s &= \textstyle\sum_{0 \le i \le j \le 2} (\text{Ind}[i = s] + \text{Ind}[j = s])\theta_{i,j} \\ b_s &= \textstyle\sum_{(i,j,k,l) \in \mathcal{S}} (\text{Ind}[i = s] + \text{Ind}[j = s] + \text{Ind}[k = s] + \text{Ind}[l = s])\rho_{i,j,k,l}. \end{aligned} \quad [11]$$

Consider a fixed triangle (u, v, w) with simplicial tie strength s. We would like to count the number of times this triangle appears in a 4-node, 4-edge subgraph configuration. Each neighbor of each of the three nodes in the triangle is either (i) the neighbor of just one node in the triangle (ii) the neighbor of exactly two nodes in the triangle, or (iii) the neighbor of all three nodes in the triangle. The first case corresponds to the induced subgraph in which we are interested, the second case to counts $\theta_{i,j}$, and the third case to counts $\rho_{i,j,k,l}$. By the inclusion-exclusion principle,

$$\psi_s = \textstyle\sum_{(u,v,w) \in \mathcal{T}_s} (d_u + d_v + d_w - 6) - 2a_s - 3b_s, \quad [12]$$

where d is the degree vector of nodes in the unweighted projected graph.

Finally, we count the 4-node subgraph configuration consisting of a triangle and an isolated node (the three leftmost configurations in the second row block of Table S4). Again, we count three types of this configuration (λ_s, $s \in \{0, 1, 2\}$), one for each of the three simplicial tie strengths of the triangle. Every triangle appears in $(n - 3)$ non-induced subgraphs with an isolated node, so we only need to subtract induced subgraph counts with more edges. We already counted these above, so the counts λ_s are given by

$$\lambda_s = |\mathcal{T}_s|(n - 3) - \psi_s - a_s - b_s. \quad [13]$$

Score functions, higher-order link prediction performance, and example predictions

We derive algorithms for higher-order link prediction, which fall into four broad categories for determining the score $s(i,j,k)$ of a triple of nodes:

1. $s(i,j,k)$ depends only on the weights of the edges (i,j), (i,k), and (j,k) in the projected graph
2. $s(i,j,k)$ is based on the local neighborhood features in the projected graph such as the common neighbors of nodes i, j, and k;
3. $s(i,j,k)$ comes from a random-walk-based similarity score
4. $s(i,j,k)$ is a learned logistic regression model in a feature-based supervised learning setting.

Several of these score functions are generalizations of traditional approaches for dyadic link prediction (13) to account for higher-order structure.

Here we introduce some notation for this section. We denote the set of simplices that node u appears in by $R(u)$; formally, $R(u) = \{S_i \mid u \in S_i\}$. The (weighted) projected graph of a dataset is the graph on node set V, where the weight of edge (u,v) is the number of simplices containing both u and v. In other words, the $|V| \times |V|$ weighted adjacency matrix W of the projected graph is defined by

$$W_{uv} = \begin{cases} |R(u) \cap R(v)| & u \neq v \\ 0 & u = v \end{cases} \qquad [14]$$

Sometimes, we will only need to consider unweighted version of the projected graph, which is encoded by the adjacency matrix A with entries $A_{uv} = \min(W_{uv}, 1)$. Finally, we denote the neighbors of a node u in the projected graph by $N(u) = \{v \in V \mid W_{uv} > 0\}$.

1. Weights in the projected graph. We use three score functions based on the weights of the pair-wise edges in the subgraph induced by nodes i, j, and k. The motivation for these methods is that weight-based tie strength positively correlates with probabilities of simplicial closure events in an aggregate sense (see the main text). Therefore, larger weights amongst the edges between nodes i, j, and k should yield larger scores. To this end, we use the following as score functions:

$$\textit{the harmonic mean:}\ s(i,j,k) = 3/(W_{ij}^{-1} + W_{ik}^{-1} + W_{jk}^{-1}) \qquad [15]$$

$$\textit{the geometric mean:}\ s(i,j,k) = (W_{ij}W_{ik}W_{jk})^{1/3} \qquad [16]$$

$$\textit{the arithmetic mean:}\ s(i,j,k) = (W_{ij} + W_{ik} + W_{jk})/3. \qquad [17]$$

As discussed in the main text, these functions are all special cases of the generalized mean function.

2. Local neighborhood features. The next set of score functions use local neighborhood features such as common neighbors of a triple of nodes. The reasoning here is that common neighborhood structure amongst a triple of nodes are positive indicators of association of the nodes; in fact, these score functions are generalizations of traditional methods used in dyadic link prediction (13). The common neighbors score function for a triple of nodes i, j, and k is the number of nodes that have appeared in at least one simplex with each of the three nodes in the candidate set:

$$\textit{3-way common neighbors:}\ s(i,j,k) = |N(i) \cap N(j) \cap N(k)|, \qquad [18]$$

where again $N(x)$ is the set of neighbors of node x in the projected graph.

The Jaccard coefficient score normalizes the number of common neighbors by the total number of neighbors of the three candidate nodes:

$$\textit{3-way Jaccard coefficient:}\ s(i,j,k) = \frac{|N(i) \cap N(j) \cap N(k)|}{|N(i) \cup N(j) \cup N(k)|}. \qquad [19]$$

This score function has been used as a general multi-way similarity measurement for binary vectors (14), but has not been employed for a link prediction task until now.

Adamic and Adar proposed log-scaled normalization for features of common neighbors between two nodes (15). Here we adapt this to a score that performs the same normalization over the common neighbors of three nodes:

$$\textit{3-way Adamic-Adar:}\ s(i,j,k) = \sum_{l \in N(i) \cap N(j) \cap N(k)} \frac{1}{\log |N(l)|}. \qquad [20]$$

Prior studies on the evolution of coauthorship have suggested preferential attachment (PA)—in terms of degree in the coauthorship network—as a mechanism for dyadic link formation (16, 17). We use two scores based on a preferential attachment model of link formation: first is

$$\textit{projected graph degree based PA:}\ s(i,j,k) = |N(i)| \cdot |N(j)| \cdot |N(k)| \qquad [21]$$

$$\textit{simplicial degree based PA:}\ s(i,j,k) = |R(i)| \cdot |R(j)| \cdot |R(k)|. \qquad [22]$$

3. Paths and random walks. The next set of scores functions are path-based metrics that ascribe higher scores when there are more paths in the projected graph between a candidate triple of nodes. Recall that A and W are the unweighted and weighted adjacency matrices for the projected graph of a dataset.

The Katz score between two nodes is the sum of geometrically damped length-l paths between two nodes (18). Katz scores have been used as a criterion for predicting dyadic links (13, 19). Formally, the Katz score between two nodes i and j in the unweighted projected graph is $\sum_{l=1}^{\infty} \beta^l A^l_{ij}$, where β is the damping parameter and A^l_{ij} counts the number of length-l paths between i and j. All pairwise Katz scores can be computed in matrix form as:

$$K^{(u)} = (I - \beta A)^{-1} - I. \qquad [23]$$

In order to guarantee that the weighted sum of length-l path lengths converges, we require that $\beta < 1/\sigma_1(A)$, the principal singular value of A (this guarantees that $I - \beta A$ is nonsingular). We chose $\beta = \frac{1}{4\sigma_1(A)}$ in our experiments.

We can also use paths in the original (weighted) projected graph, where W^l_{ij} is the number of length-l paths between i and j if we interpret the integer weights in W to be parallel edges. This leads to the weighted pairwise Katz scores

$$K^{(w)} = (I - \beta W)^{-1} - I. \qquad [24]$$

Again, β must be less than $1/\sigma_1(W)$ to guarantee that $(I - \beta W)$ is nonsingular, and we choose $\beta = \frac{1}{4\sigma_1(W)}$ in our experiments.

Given the pairwise Katz scores, we define score functions for triples of nodes as follows:

$$\textit{unweighted 3-way Katz: } s(i,j,k) = K^{(u)}_{ij} + K^{(u)}_{ik} + K^{(u)}_{jk} \qquad [25]$$

$$\textit{weighted 3-way Katz: } s(i,j,k) = K^{(w)}_{ij} + K^{(w)}_{ik} + K^{(w)}_{jk}. \qquad [26]$$

For many of our datasets, storing the K matrices in a dense format requires too much memory. In these cases, we use the Krylov subspace method MINRES (20) to solve the linear systems

$$(I - \beta A)k_j = e_j, \quad j = 1, \ldots, |V|, \qquad [27]$$

where e_j is the jth standard basis vector. After computing k_j, we store only the entries of the jth column of K corresponding to the sparsity pattern of the jth column of A. These are the only entries of K needed for computing the scores in Eq. (25).

The personalized PageRank (PPR) score is another path-based score used in dyadic link prediction (13, 21). PPR is based on the random walk underlying the classical PageRank ranking system for web pages (22). More specifically, consider a Markov chain, where at each step, with probability $0 < \alpha < 1$, the chain transitions according to a random walk in a graph, and with probability $1 - \alpha$ transitions to node i. The PPR score of node j with respect to node i is then the stationary probability of the state j for the Markov chain. The PPR scores are given by the matrix

$$F^{(u)} = (1 - \alpha)(I - \alpha A D^{-1})^{-1}, \qquad [28]$$

where $F^{(u)}_{ji}$ is the PPR score of j with respect to node i. Here D is the diagonal degree matrix, $D_{jj} = \sum_i A_{ij}$. We can again provide an analog for the weighted case:

$$F^{(w)} = (1 - \alpha)(I - \alpha W D_W^{-1})^{-1}, \qquad [29]$$

where $[D_W]_{jj} = \sum_i W_{ij}$ is the weighted diagonal degree matrix.

As we did with the Katz scores, we construct three-way scores from the pairwise PPR scores:

$$\textit{unweighted 3-way PPR: } s(i,j,k) = F^{(u)}_{ij} + F^{(u)}_{ji} + F^{(u)}_{ik} + F^{(u)}_{ki} + F^{(u)}_{jk} + F^{(u)}_{kj} \qquad [30]$$

$$\textit{weighted 3-way PPR: } s(i,j,k) = F^{(w)}_{ij} + F^{(w)}_{ji} + F^{(w)}_{ik} + F^{(w)}_{ki} + F^{(w)}_{jk} + F^{(w)}_{kj}. \qquad [31]$$

(Unlike the Katz score matrices K, the PPR matrices are not symmetric, so we account for both directions of the edges.)

We also use a recent generalization of PPR scores for abstract simplicial complexes, based on tools from algebraic topology (23). Here, we describe the computations necessary for these scores, assuming a basic knowledge of algebraic topology.

We consider the abstract simplicial complex defined by the union of the set of closed triangles T, the set of edges E, and the set of vertices V. We orient the edges and triangles so that (i, j) for $i < j$ corresponds to an edge $\{i, j\}$ and (i, j, k) for $i < j < k$ corresponds to a closed triangle $\{i, j, k\}$. Following the ideas of Schaub et al., we define the normalized combinatorial Hodge Laplacian as

$$\hat{\Delta} = (G D^{-1} G^T + C^T C) M^{-1}, \qquad [32]$$

where the "gradient operator" G is a $|E| \times |V|$ matrix defined by

$$G_{(i,j),x} = \begin{cases} 1 & x = j \\ -1 & x = i \\ 0 & \text{otherwise,} \end{cases} \qquad [33]$$

the "curl operator" C is a $|T| \times |E|$ matrix defined by

$$C_{(i,j,k),(x,y)} = \begin{cases} 1 & (x,y) = (i,j) \text{ or } (x,y) = (j,k) \\ -1 & (x,y) = (i,k) \\ 0 & \text{otherwise,} \end{cases} \qquad [34]$$

D is a diagonal matrix defined by

$$D_{xx} = \textstyle\sum_{(i,j)} |G_{(i,j),x}|, \qquad [35]$$

and M is a diagonal matrix defined by

$$M_{(x,y),(x,y)} = 2 + \textstyle\sum_{(i,j,k)} |C_{(i,j,k),(x,y)}|. \qquad [36]$$

The matrix $P = \frac{1}{2}(I - \hat{\Delta})$ defines a Markov-like operator. The simplicial PageRank scores (defined on each pair of edges) can thus be defined analogously to the standard PageRank:

$$S = (I - \alpha P)^{-1}(1 - \alpha). \qquad [37]$$

Here, the matrix S defines pairwise scores between *edges*, and we construct a score function on triples of nodes by taking the sum of pairwise scores:

3-way simplicial PPR:

$$s(i,j,k) = |S_{(i,j),(j,k)}| + |S_{(j,k),(i,j)}| + |S_{(i,j),(i,k)}| + |S_{(i,k),(i,j)}| + |S_{(j,k),(i,k)}| + |S_{(j,k),(i,k)}|. \qquad [38]$$

4. Supervised learning. Finally, we used a supervised machine learning approach that learns the appropriate score function given features of the open triangle. To this end, we further divide the training data into a sub-training set (simplices appearing in the first 60% of the entire dataset) and a validation set (simplices appearing between the 60th and 80th percentile of the time spanned by the entire dataset). We trained an ℓ_2-regularized logistic regression model using the scikit learn library[‡‡] (24) for predicting closure on the validation set using features of open structures in the sub-training set. The features for each open triangle (i,j,k) were

1. the number of simplices containing pairs of nodes i and j, i and k, and j and k;
2. the degree of nodes i, j, and k in the projected graph: $|N(i)|$, $|N(j)|$, and $|N(k)|$;
3. the number of simplices containing nodes i, j, and k: $|R(i)|$, $|R(j)|$, and $|R(k)|$;
4. the number of common neighbors in the projected graph of nodes i and j, i and k, and j and k: $|N(i) \cap N(j)|$, $|N(i) \cap N(k)|$, and $|N(j) \cap N(k)|$;
5. the number of common neighbors of all three nodes i, j, and k in the projected graph: $|N(i) \cap N(j) \cap N(k)|$
6. the log of the features in Items 1 to 3 and the log of the sum of 1 and the feature value for the features in Items 4 and 5.

After learning the model, we predicted on the test set using the same features computed on the entire training set (first 80% of the dataset).

Prediction performance. Using the ranking induced by the score functions described above, we evaluated the prediction performance on each dataset by the area under the precision-recall curve (AUC-PR) metric (Table S5). We use random scores—more specifically, a random ranking—as a baseline, and report scores relative to this baseline.

As seen in the main text, our proposed algorithms can achieve much higher performance than randomly guessing which open triangles go through a simplicial closure event. We also still see good performance of the harmonic and geometric means, as well as the supervised problem, with respect to this expanded set of score functions.

We may further decompose the pairwise scores of simplicial PageRank scores in Eq. (38) into the gradient, harmonic, and curl components given by the Hodge decomposition (23). Computationally, we solve the least squares problems

$$\min_X \|GX - S\|_F, \qquad \min_Y \|C^T Y - S\|_F \qquad [39]$$

using the iterative method LSQR (25) (with tolerances 10^{-3}) on each column. Given the minimizers X^* and Y^* of Eq. (39), the components of the Hodge decomposition are

$$S_{\text{grad}} = GX^*, \qquad S_{\text{curl}} = C^T Y^*, \qquad S_{\text{harm}} = S - S_{\text{grad}} - S_{\text{curl}}. \qquad [40]$$

Each of S_{grad}, S_{curl}, and S_{harm} defines pairwise scores between edges, and we construct score functions on triples of nodes in the same way as in Eq. (38).

We report the performance results in Table S6 for the datasets that were small enough on which computing the Hodge decomposition was computationally feasible. We observe that the components from the Hodge decomposition can provide substantially better results than the "combined" simplicial PageRank score reported Table S5. However, no component consistently out-performs the others.

[‡‡] http://scikit-learn.org/

Table S5. Open triangle closure prediction performance based on several score functions: random (Rand.); harmonic, geometric, and arithmetic means of the 3 edge weights; 3-way common neighbors (Common); 3-way Jaccard coefficient (Jaccard); 3-way Adamic-Adar (A-A); projected graph degree and simplicial degree preferential attachment (PGD-PA and SD-PA); unweighted and weighted Katz similarity (U-Katz and W-Katz); unweighted and weighted personalized PageRank (U-PPR and W-PPR); simplicial personalized PageRank (S-PPR; missing entries are cases where computations did not finish within 2 weeks); and a feature-based supervised model using logistic regression (Log. reg.). Performance is AUC-PR relative to the random baseline. The random baseline is listed in absolute terms and equals the fraction of open triangles that close. The harmonic and geometric means of edge weights perform well across many datasets, further highlighting the role of tie strength in predicting simplicial closure events. This signal from local structure contrasts from traditional pairwise link prediction, where longer paths are needed for effective prediction (13). The supervised method also performs well, suggesting that combinations of features capture the rich variety of structure observed across datasets.

Dataset	Rand.	Harm. mean	Geom. mean	Arith. mean	Common	Jaccard	A-A	PGD-PA	SD-PA	U-Katz	W-Katz	U-PPR	W-PPR	S-PPR	Log. reg.
coauth-DBLP	1.68e-03	1.49	1.59	1.50	1.33	1.84	1.60	0.74	0.74	0.97	1.51	1.62	1.83	1.21	3.37
coauth-MAG-History	7.16e-04	1.69	2.72	3.20	5.11	2.24	5.82	1.50	2.49	6.30	3.40	1.66	1.88	1.35	6.75
coauth-MAG-Geology	3.35e-03	2.01	1.97	1.69	2.43	1.84	2.71	1.31	0.97	1.99	1.74	1.06	1.26	0.94	4.74
music-rap-genius	6.82e-04	5.44	6.92	1.98	1.85	1.62	2.10	1.82	2.15	1.93	2.00	1.78	2.09	1.39	2.67
tags-stack-overflow	1.84e-04	13.08	10.42	3.97	6.45	9.43	6.63	3.37	2.74	2.95	3.60	1.08	1.85	–	3.37
tags-math-sx	1.08e-03	9.08	8.67	2.88	6.19	9.37	6.34	3.48	2.81	4.53	2.71	1.19	1.55	1.86	13.99
tags-ask-ubuntu	1.08e-03	12.29	12.64	4.24	7.15	4.96	7.51	7.48	5.63	7.10	4.15	1.75	2.54	1.19	7.48
threads-stack-overflow	1.14e-05	23.85	31.12	12.97	2.73	3.85	3.19	5.20	3.89	1.06	11.54	1.66	4.06	–	1.53
threads-math-sx	5.63e-05	20.86	16.01	5.03	25.08	28.13	23.32	10.46	7.46	11.04	4.86	0.90	1.18	0.61	47.18
threads-ask-ubuntu	1.31e-04	78.12	80.94	29.00	21.04	2.80	30.82	7.09	6.62	16.63	32.31	0.94	1.51	1.78	9.82
NDC-substances	1.17e-03	4.90	5.27	2.90	5.92	3.36	5.97	4.76	4.46	5.35	2.93	1.39	1.83	1.86	8.17
NDC-classes	6.72e-03	4.43	3.38	1.82	1.27	1.19	0.99	0.94	2.14	0.92	1.34	0.78	0.91	2.45	0.62
DAWN	8.47e-03	4.43	3.86	2.13	4.73	3.76	4.77	3.76	1.45	4.61	2.04	1.57	1.37	1.55	2.86
congress-committees	6.99e-04	3.59	3.28	2.48	4.83	2.49	5.04	1.06	1.31	3.21	2.59	1.50	3.89	2.13	7.67
congress-bills	1.71e-04	0.93	0.90	0.88	0.65	1.23	0.66	0.60	0.55	0.60	0.78	3.16	1.07	6.01	107.19
email-Enron	1.40e-02	1.78	1.62	1.33	0.85	0.83	0.87	1.27	0.83	0.99	1.28	3.69	3.16	2.02	0.72
email-Eu	5.34e-03	1.98	2.15	1.78	1.28	2.69	1.37	0.88	1.55	1.01	1.79	1.59	1.75	1.26	3.47
contact-high-school	2.47e-03	3.86	4.16	2.54	1.92	3.61	2.00	0.96	1.13	1.72	2.53	1.39	2.41	0.78	2.86
contact-primary-school	2.59e-03	5.63	6.40	3.96	2.98	2.95	3.21	0.92	0.94	1.63	4.02	1.41	4.31	0.93	6.91

Table S6. Open triangle closure prediction performance based on score functions from the Hodge decomposition of the simplicial personalized PageRank vector.

Dataset	Rand.	combined	gradient	harmonic	curl
coauth-MAG-History	7.16e-04	1.35	1.25	1.13	1.27
music-rap-genius	6.82e-04	1.39	1.44	1.40	1.47
tags-math-sx	1.08e-03	1.86	0.73	0.66	0.74
tags-ask-ubuntu	1.08e-03	1.19	0.61	0.59	0.71
threads-ask-ubuntu	1.31e-04	0.61	0.58	0.61	4.59
NDC-substances	1.17e-03	1.86	0.63	0.72	0.60
NDC-classes	6.72e-03	2.45	1.37	0.83	1.74
DAWN	8.47e-03	1.55	0.59	0.60	0.65
congress-committees	6.99e-04	2.13	1.22	1.13	1.63
email-Enron	1.40e-02	2.02	2.90	1.98	2.46
email-Eu	5.34e-03	1.26	1.28	0.82	1.63
contact-high-school	2.47e-03	0.78	0.99	1.68	2.38
contact-primary-school	2.59e-03	0.93	1.45	1.84	3.26

Table S7. Top 25 predictions from the 3-way Adamic-Adar algorithm for open triangles to go through a simplicial closure event in the DAWN dataset. An “X” marks open triangles that actually go through a simplicial closure event final 20% of the time spanned by the dataset. Four of the top 25 predictions do indeed have a simplicial closure event.

1		methyldopa; gentamicin; proton pump inhibitors
2	X	norepinephrine; chlormezanone; proton pump inhibitors
3		ranitidine; gentamicin; proton pump inhibitors
4		dihydroergotamine; methyldopa; asa/butalbital/caffeine/codeine
5		ranitidine; gentamicin; levodopa
6		praziquantel; diazepam; alfentanil
7		asa/caffeine/dihydrocodeine; praziquantel; proton pump inhibitors
8		chloral hydrate; tobramycin; sumatriptan
9		oxybutynin; gentamicin; tobramycin
10		asa/caffeine/dihydrocodeine; norepinephrine; sumatriptan
11		ampicillin; chlormezanone; proton pump inhibitors
12		bepridil; diazepam; alfentanil
13		colestipol; oxybutynin; proton pump inhibitors
14	X	nadolol; benazepril; proton pump inhibitors
15		thalidomide; amiloride; maprotiline
16	X	nadolol; lamivudine-zidovudine; proton pump inhibitors
17		chloral hydrate; verapamil; methyldopa
18		chlorzoxazone; benazepril; proton pump inhibitors
19		heparin; asa/caffeine/dihydrocodeine; proton pump inhibitors
20		oxcarbazepine; norepinephrine; proton pump inhibitors
21		dihydroergotamine; tobramycin; alfentanil
22		maprotiline; norepinephrine; proton pump inhibitors
23		oxybutynin; methyldopa; dihydroergotamine
24		heparin; dihydroergotamine; proton pump inhibitors
25	X	ampicillin; methyldopa; diazepam

Example predictions. Lastly, we provide a concrete example of predictions. Table S7 shows the top 25 predictions of the Adamic-Adar score function on the DAWN dataset. In this dataset, fewer than one in a hundred open triangles in the training set experience a simplicial closure event in the test set, but 4 of the top 25 predictions from this score function go through a simplicial closure event. Three of the correct predictions relate to novel combinations with proton pump inhibitors.

References

1. Sinha A, et al. (2015) An overview of microsoft academic service (MAS) and applications in *Proceedings of the 24th International Conference on World Wide Web.* (ACM Press), pp. 243–246.
2. Porter MA, Mucha PJ, Newman MEJ, Warmbrand CM (2005) A network analysis of committees in the U.S. House of Representatives. *Proceedings of the National Academy of Sciences* 102(20):7057–7062.
3. Porter MA, Mucha PJ, Newman M, Friend A (2007) Community structure in the United States House of Representatives. *Physica A: Statistical Mechanics and its Applications* 386(1):414–438.
4. Fowler JH (2006) Legislative cosponsorship networks in the US house and senate. *Social Networks* 28(4):454–465.
5. Fowler JH (2006) Connecting the Congress: A study of cosponsorship networks. *Political Analysis* 14(04):456–487.
6. Klimt B, Yang Y (2004) The enron corpus: A new dataset for email classification research in *Machine Learning: ECML 2004.* (Springer Berlin Heidelberg), pp. 217–226.
7. Paranjape A, Benson AR, Leskovec J (2017) Motifs in temporal networks in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* (ACM Press), pp. 601–610.
8. Mastrandrea R, Fournet J, Barrat A (2015) Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLOS ONE* 10(9):e0136497.
9. Stehlé J, et al. (2011) High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE* 6(8):e23176.
10. Pinar A, Seshadhri C, Vishal V (2017) ESCAPE: Efficiently Counting All 5-Vertex Subgraphs in *Proceedings of the 26th International Conference on World Wide Web.* (ACM Press), pp. 1431–1440.
11. Latapy M (2008) Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theoretical Computer Science* 407(1-3):458–473.
12. Chiba N, Nishizeki T (1985) Arboricity and subgraph listing algorithms. *SIAM Journal on Computing* 14(1):210–223.
13. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology* 58(7):1019–1031.
14. Heiser WJ, Bennani M (1997) Triadic distance models: Axiomatization and least squares representation. *Journal of Mathematical Psychology* 41(2):189–206.
15. Adamic LA, Adar E (2003) Friends and neighbors on the web. *Social Networks* 25(3):211–230.

16. Newman MEJ (2001) Clustering and preferential attachment in growing networks. *Physical Review E* 64(2).
17. Barabási A, et al. (2002) Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications* 311(3-4):590–614.
18. Katz L (1953) A new status index derived from sociometric analysis. *Psychometrika* 18(1):39–43.
19. Wang C, Satuluri V, Parthasarathy S (2007) Local probabilistic models for link prediction in *Seventh IEEE International Conference on Data Mining*. (IEEE).
20. Paige CC, Saunders MA (1975) Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis* 12(4):617–629.
21. Bahmani B, Chowdhury A, Goel A (2010) Fast incremental and personalized PageRank. *Proceedings of the VLDB Endowment* 4(3):173–184.
22. Page L, Brin S, Motwani R, Winograd T (1999) The PageRank citation ranking: Bringing order to the web, (Stanford University), Technical Report 1999-66.
23. Schaub MT, Benson AR, Horn P, Lippner G, Jadbabaie A (2018) Random walks on simplicial complexes and the normalized hodge laplacian. *arXiv preprint arXiv:1807.05044*.
24. Pedregosa F, et al. (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.
25. Paige CC, Saunders MA (1982) LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software* 8(1):43–71.