Supplementary Information: Knowledge gaps in the early growth of semantic feature networks

Ann E. Sizemore[1], Elisabeth A. Karuza[2], Chad Giusti[3], and Danielle S. Bassett[1,4,5,6,*]

[1]Department of Bioengineering, School of Engineering and Applied Sciences, University of Pennsylvania, Philadelphia, PA 19104 USA
[2]Department of Psychology, College of Arts and Sciences, University of Pennsylvania, Philadelphia, PA 19104 USA
[3]Department of Mathematical Sciences, University of Delaware, DE 19716 USA
[4]Department of Physics & Astronomy, College of Arts and Sciences, University of Pennsylvania, PA 19104 USA
[5]Department of Neurology, Perelman School of Medicine, University of Pennsylvania, PA 19104 USA
[6]Department of Electrical & Systems Engineering, School of Engineering and Applied Sciences, University of Pennsylvania, PA 19104 USA

*dsb@seas.upenn.edu

Supplementary Methods

Details of topological methods

This section outlines the details of encoding the growing semantic feature network as a filtration and computing the persistent homology. We devote significant real estate to the encoding and briefly describe persistent homology since it is more thoroughly discussed elsewhere.[13,14,67,72]

Node-weighted networks and induced filtrations

Our motivation comes from data described most naturally as a network with weights on the nodes. Such systems can arise from protein-protein interaction networks with protein expression as node weights, structural brain networks with region activity as node weights, or a social network with time of contamination as nodes weights. Though generally classic graph statistics do not extend easily to networks with node weights, we present a construction that allows the simultaneous computation of both graph statistics and persistent homology on node-weighted networks.

It is important to note that we derive our inspiration from an object generated from edge-weighted networks called the *order complex*.[10] Given a graph with edge weights, we obtain an ordering on the edges by decreasing edge weights. Then we create a sequence of graphs, $G_0 \subset G_1 \subset \dots G_{|E|}$ with each G_i the graph containing the i highest ranked edges. This sequence of graphs is called the *order complex* of the weighted network (or corresponding symmetric weighted matrix).

Now returning to node-weighted networks, we get a node ordering from the decreasing node vales. Then from this node ordering s and graph G with N nodes we can similarly construct a sequence of graphs $G_0 \subset G_1 \subset \cdots \subset G_N$ with G_n containing the first n nodes in s and any connections between these nodes which exist in G. We call this sequence of binary graphs the *node-filtered order complex* of a node-weighted network, denoted $\text{nord}(G, w)$ with G the binary graph and $w : N \to \mathbb{R}$ the function assigning node weights. For brevity we often refer to this object as the n-order complex. Note then the n-order complex is completely determined by the pair (G, w) or (G, s) with s the node ordering.

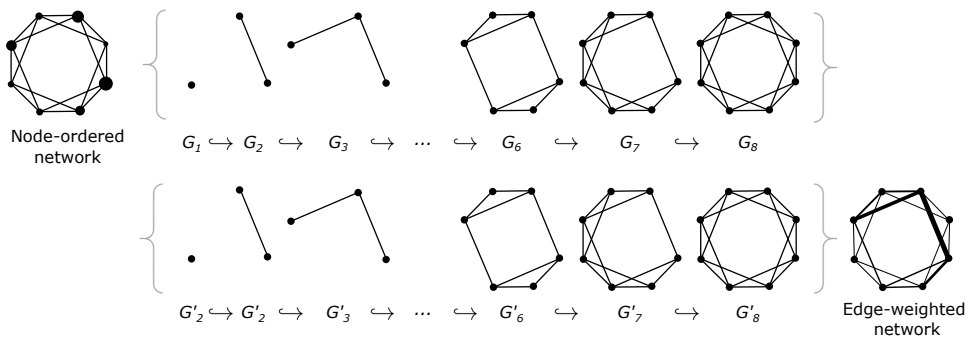
The n-order complex retains intrinsic developmental aspects of the node-weighted system, and furthermore allows for computation of both common binary graph metrics and persistent homology on these objects. Most graph metrics are not generalizable to include orders (weights) on the nodes, but if we instead construct the n-order complex, we now can compute such metrics on each G_n in the filtration.

Construction of n-order complex into order complex

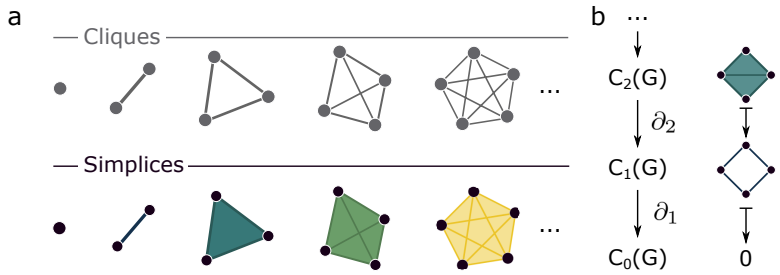
In practice, the software that we use to compute the persistent homology expects oder complexes. Then to compute the persistent homology of a growing network, we encode the associated n-order complex as an order complex. We will show by construction that any n-order complex $\text{nord}((G, s))$ can be translated into a weighted network M with each G_i in the filtration of $\text{nord}((G, s))$ equal to G'_i in the filtration of $\text{ord}(M)$.

Given a node-ordered network (G, s), we construct the node-filtered order complex $\text{nord}((G, s))$. Then each G_i is a binary graph and can be written as an $N \times N$ binary symmetric matrix M_i (with $N - n$ extra padding rows and columns). Let $M = \sum_{i=0}^{N} M_i$. Now we have created a real-valued symmetric matrix M that encodes a weighted network with the highest edge weights corresponding to the earliest added edges. If we then create $\text{ord}(M)$ with binary graphs G'_i, then by our construction it must be that $G'_i = G_i$ for all $i = 0, \dots, N$. Thus any n-order complex can be written as a weighted matrix M with $\text{ord}(M) = \text{nord}((G, s))$. It is important to note that to get $G_i = G'_i$, we must assume that either all nodes always exist and we use the "growing graph" concept to describe how edges are added, or for both we do not include nodes in the graph until they have a neighbor. Finally, if we denote by $\mathcal{N}$ the set of filtrations achievable by node-weighted networks and $\mathcal{O}$ the set of filtrations achievable by edge-weighted networks, by the above discussion we must have $\mathcal{N} \subseteq \mathcal{O}$.

To examine the reverse relation, take the 3-clique with nodes a, b, and c with weights $e_{a,b} = 3$, $e_{b,c} = 2$, $e_{a,c} = 1$. The resulting filtration could not be created from a node-weighted network, since the final edge



Supplementary Figure 1: **Filtration of a node-ordered network (top) and equivalent edge-weighted network yielding the same graph filtration (bottom).**



Supplementary Figure 2: **Simplices and boundaries.** *(a)* A $(k+1)$-clique is a collection of $k+1$ all-to-all connected nodes (1-clique, 2-clique, etc. shown in top). Cliques in a graph are replaced with simplices of the same number of nodes (0-simplex, 1-simplex, etc. shown at bottom). *(b)* Example of an element in $C_2(G)$ sent to its boundary in $C_1(G)$, then when the boundary is again taken this is sent to 0 in $C_0(G)$.

$e_{a,c}$ would be added to two nodes that already exist. In other words, it could not be added as the result of adding a new node. Therefore $\mathcal{N} \subsetneq \mathcal{O}$.

Persistent homology

Next we formally describe persistent homology. We begin with the task of detecting topological cavities in an unweighted graph. Given $G = (V, E)$ we translate this binary, unweighted graph into a combinatorial object called the clique complex by "coloring in" all cliques (all-to-all connected subgraphs) of G. Formally every $(k+1)$-clique, a completely connected subgraph of G containing $k+1$ nodes (Supplementary Fig. 2a, top), is replaced with a k-simplex (Supplementary Fig. 2a, bottom). A k-simplex $\sigma = \{v_0, v_1, \ldots, v_k\}$ is the convex hull of $k+1$ affinely positioned nodes. The collection of simplices created from the cliques of G is called the *clique complex* $X(G)$. The clique complex of G is an abstract simplicial complex, meaning that we have a vertex set V (the original vertex set of G) along with a collection K of subsets of V that is closed under taking subsets. So elements of K are simplices, and using geometric intuition we see clearly that any subset of a simplex must also be a simplex, called a *face*. We can write $X(G) = \{X_0(G), X_1(G), \ldots, X_M(G)\}$ with each $X_k(G)$ being the collection of k-simplices of G called the k-skeleton. To summarize thus far we have taken our graph G and translated it into the combinatorial object called the clique complex $X(G)$.

To locate topological cavities, we will need to perform algebra with elements of the clique complex. We create the *chain group* $C_k(X(G))$, a vector space with basis elements σ corresponding to k-simplices of $X(G)$. Then elements of $C_k(X(G))$, called k-chains, are linear combinations of these basis elements. Though we can certainly choose coefficients from any group (for example, $\mathbb{Z}$), for computational purposes we work in the field $\mathbb{Z}_2$. To streamline notation, we will write $C_i(G)$ to mean $C_i(X(G))$ for simplicity.

Boundary operator. To locate the topological cavities, we need to first comprehend the makeup and arrangements of simplices in our simplicial complex. For example, if we only have edges, we cannot tell which closed loops are true cavities without information about the positions of higher dimensional simplices within the complex. In particular, when searching for k-dimensional cavities we need to know the k-dimensional footprints of $(k+1)$-dimensional simplices. These footprints are the boundaries of $(k+1)$-simplices that can be computed using the boundary operator ∂_{k+1}. The boundary operator $\partial_{k+1} : C_{k+1} \to C_k$ is defined

$$\partial_{k+1}(\sigma_{v_0,v_1,\ldots,v_k}) = \sum_i (-1)^i \sigma_{v_0,v_1,\ldots,v_{i-1},\hat{v_i},v_{i+1},\ldots,v_k}$$

with $\hat{v_i}$ omitted. Note the $(-1)^i$ records the directionality of chains, but since we work in $\mathbb{Z}_2$ we can drop this term.

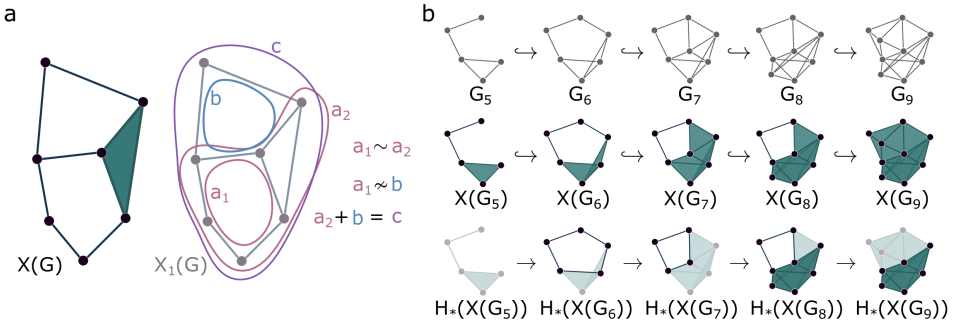
The boundary operator allows us to detect cavities due to a few particularly useful properties. First, the boundary operator extends linearly, so for $a, b \in C_k$, $\partial_k(a+b) = \partial_k(a) + \partial_k(b)$. Geometrically this means that the boundary of a collection of simplices is what we would intuit: the $(k-1)$-simplices that form a "shell" around the k-chain $a+b$ (for example, see Supplementary Fig. 2b).

Next, let us examine what happens when we take the boundary of a cycle, a closed path of simplices. Again following geometric intuition, the boundary of a cycle is the end minus the beginning, which are the same in a closed path, and therefore the boundary should be 0. Indeed, cycles of dimension k are precisely the elements in C_k sent to 0 by ∂_k, or $\ker(\partial_k)$. Now note that one way we could construct a cycle is to take a $(k+1)$-simplex σ and remove the interior – equivalently send σ to its k-boundary. Then $\partial_{k+1}(\sigma)$ is a cycle, and thus $\partial_k(\partial_{k+1}(\sigma)) = 0$. If the boundary of any simplex is a cycle, then by linearity we get that the boundary of any $(k+1)$-chain is a k-cycle. Thus $\mathrm{im}(\partial_{k+1}) \subseteq \ker(\partial_k)$. We call elements of $\mathrm{im}(\partial_{k+1})$ k-boundaries. To summarize, we have $\ker(\partial_k)$ the k-cycles, $\mathrm{im}(\partial_{k+1})$ the k-boundaries, and $\partial_k \circ \partial_{k+1} = 0$ (so $\mathrm{im}(\partial_{k+1}) \subseteq \ker(\partial_k)$).

Equivalent cycles. We have just seen how all k-boundaries are necessarily k-cycles. But what if $\mathrm{im}(\partial_{k+1}) \subsetneq \ker(\partial_k)$? Then there exist k-cycles in $\ker(\partial_k) - \mathrm{im}(\partial_{k+1})$ that do not surround a collection of higher dimensional simplices. Thus, they must instead enclose a void of dimension $k+1$ called a k-cavity. Since the cavities themselves are the features of interest, we do not want to simply enumerate $\ker(\partial_k) - \mathrm{im}(\partial_{k+1})$, but instead we need to have all cycles surrounding the same cavity count as one (to avoid grossly overcounting). If two cycles surround the same cavity (we will assume each only surrounds one cavity for the sake of this example) then their difference must be some collection of higher dimensional simplices. More precisely, if we let ℓ_1 and ℓ_2 denote these two k-cycles, then $\ell_1 - \ell_2 \in \mathrm{im}(\partial_{k+1})$. We call these two cycles *equivalent*. In fact, we say that any two cycles $a, b \in \ker(\partial_k)$ are equivalent if $a - b \in \mathrm{im}(\partial_{k+1})$. For example, we see in Supplementary Fig. 3a the two cycles a_1 and a_2 are equivalent because $a_2 - a_1$ is the boundary of a 2-simplex. However, $a_1 \not\sim b$, since their difference is not a boundary of a collection of 2-simplices. We could also take the 1-cycle c which surrounds both cavities, though note that it is not equivalent to any of a_1, a_2, or b but is instead the sum $a_2 + b \sim a_1 + b$. The defined equivalence relation partitions $\ker(\partial_k)$ into equivalence classes $[\ell_0] = \{\ell \in Z_k | \ell_0 - \ell \in \mathrm{im}(\partial_{k+1})\}$. Then each (non-trivial) equivalence class corresponds to a topological cavity within the simplicial complex. By abuse it is common to refer to an equivalence class of k-cycles as a k-cycle.

Homology. At this point in the exposition, we have detailed the intuitions and definitions necessary to concretely define the homology groups of simplicial complexes. The homology group is simply the group formed by the equivalence classes as we defined above. Formally stated $H_k(X(G)) := \ker(\partial_k)/\mathrm{im}(\partial_{k+1})$. Each non-trivial equivalence class corresponds to a topological cavity, so $\dim(H_k(X(G)))$ is the number of k-cavities within the simplicial complex X. The dimension of $H_k(X(G))$ is called the k^{th} Betti number β_k and the list $\{\beta_0, \beta_1, \ldots, \beta_m\}$ are the *Betti numbers* of $X(G)$.

Revisiting filtrations. Earlier we introduced filtrations as a way to encode node-weighted networks. Consider



Supplementary Figure 3: **Equivalent cycles and filtrations.** *(a)* For the clique complex $X(G)$, 1-cycles a_1 and a_2 are equivalent since their difference is a boundary of a 2-simplex. However, these cycles are not equivalent to either cycle b or c. *(b)* Filtration on graphs G_i (top) induces a filtration on their clique complexes $X(G_i)$ (middle) which finally induces maps between homology groups $H_*(X(G_i))$ (bottom). The minimal 1-cycle surrounding a cavity born at node 6 is highlighted as it shrinks and dies at node 9. A minimal 2-cycle born at node 8 is also highlighted.

one unit of the filtration, the map $i_k : G_k \hookrightarrow G_{k+1}$. Since every node and edge in G_k maps to itself in G_{k+1}, we see that the map i extends to clique complexes, with every simplex in $X(G_k)$ mapping to itself in $X(G_{k+1})$. This gives us the map $i'_k : X(G_k) \hookrightarrow X(G_{k+1})$. Then, the filtration of graphs induces a filtration of clique complexes (Supplementary Fig. 3b). Furthermore, the inclusion $X(G_k) \hookrightarrow X(G_{k+1})$ also means that we can easily map elements of the chain groups $C_*(X(G_k)) \hookrightarrow C_*(X(G_{k+1}))$, since we can take the above inclusion i'_k as mapping the basis elements of $C_*(X(G_k))$ to those in $C_*(X(G_{k+1}))$. We depict these concepts in Supplementary Fig. 3b.

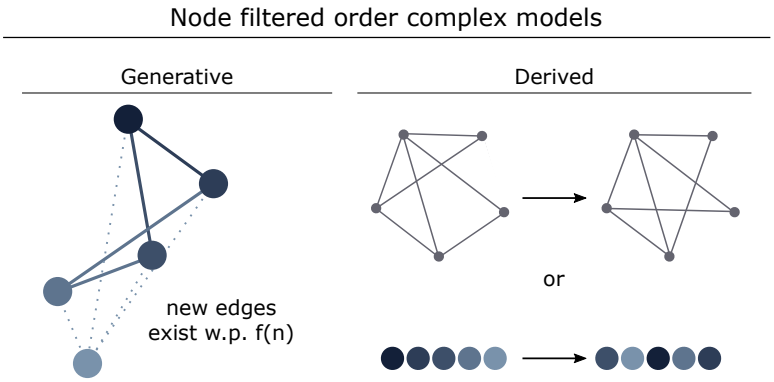
Finally since we have these nice maps from one chain complex into the next, we can map cycles to cycles and consequentially the homology groups $H_*(X_k) \to H_*(X_{k+1})$ (Supplementary Fig. 3b, bottom). This means we can not only find k-cavities at each filtration index, but we can *follow* each k-cavity from the first point it exists in the filtration (called the birth), as it evolves throughout the filtration, and is killed (called the death) by simplices tessellating the cavity. Some cavities never die, so we assign them a death time of inf. We call the *lifetime* of a persistent cycle the $death - birth$. The birth and death can be given in terms of the edge density,[10,73] filtration index, or for this study the number of nodes added. For example, the persistent 1-cycle in Supplementary Fig. 3b is born with the addition of node 6 and dies when node 9 is added, resulting in a lifetime = 3.

Models of node-filtered order complexes

While this exposition is motivated by early semantic learning, creating simple models with controlled properties will help us gain an intuition for possible behaviors of n-order complexes. Though we describe the following models in the main text, for conceptual organization we revisit the definitions and group by model type.

We describe two main categories of n-order complex models: generative and derived (Supplementary Fig. 4). Recall a n-order complex can be completely defined by the pair (G, s) with G a binary graph and s an ordering of the nodes. A generative model creates the complex according to a set of rules: for n-order complexes we can generate a pair (G, s) by first assuming $s = 1, 2, \ldots, N$ and then constructing G. Derived models instead begin with either G or s and use this to construct the model.

We can construct simple n-order complex models using a function $p : \{1, 2, \ldots, N\} \to [0, 1]$ such that when node n is added, each edge between node n and all previous nodes exist with probability $p(n)$. In the main text we include results for



Supplementary Figure 4: **Models for n-order complexes.** We design models falling into two groups: Generative and Derived. Those in the generative group are created by assigning probabilities to edge presence with each node added. Derived models can be constructed from existing complexes by altering the edges in the binary graph or by altering the order of the nodes.

$$p(n) = c, \tag{3}$$

$$p(n) = (n/N)^d, \tag{4}$$

called the constant probability model and proportional probability model, respectively.

We could also enforce some global architecture such as community structure on the graph. If we let c_n be the community of node n, then we can iteratively build a binary graph G at step n

$$p(n,m) = \left\{ \begin{array}{ll} p_{in} & c_n = c_m; \\ p_{out} & c_n \neq c_m \end{array} \right\} \tag{5}$$

with $p_{in} > p_{out}$ defining within or between module edge probabilities and $m < n$. We assign the community affiliation vector randomly with the desired number of communities and call this the modular n-order complex model.

Instead of constructing a graph with a particular global structure, it may be the case that we may have some local information such as a predetermined affinity of each node for connections. The node affinity does not change as the network grows, making this inherently different than for example the preferential attachment model.[74] Then given an affinity vector $a = (a_1, a_2, \ldots, a_N)$ with a_m the affinity of node m, we can construct a n-order complex using the following rule: when the nth node is added, the probability of an edge forming between node n, m is

$$p(n,m) = c\frac{a_m}{\max(a)} \ . \tag{6}$$

We call this the edge affinity model. If a node with normalized affinity = 1 is not ideal, one can multiply $\frac{a_m}{\max(a)}$ by a constant c to adjust the maximum probability that any node will acquire edges after it is added.

The second class of models that we consider we call derived models, which, in contrast to generative models, alter features of an existing network and therefore require some prior knowledge of the system. We further group these into two basic types based on whether the edges of the binary graph G or the node ordering s changes. In the first, we maintain the original node ordering s but, as an example, could randomly rewire the edges of G while preserving degree distribution (similar to the configuration model[19,20]) which we call the randomized edges model. In the second, we maintain the original graph G and reorder the nodes, either randomly (randomized node model) or perhaps based on a node property of G such as degree (decreasing degree model) or topological distance from a given node (distance from v_0 model).

Additional information for null models

Parameters for the four presented generative n-order complex models were chosen to match the edge density of the semantic feature network (~ 0.3). One might ask how these parameters affect the persistent homology of the growing graphs. In Supplementary Fig 5 we show the persistent homology of the constant probability model with $p = 0.2$ and $p = 0.4$, the proportional probability model with $d = 0.5$ or $d = 2$, the modular model with $p_{in} = 0.8$, $p_{out} = 0.2$ and $p_{in} = 0.6$, $p_{out} = 0.4$, and the edge affinity model with affinity vectors as random permutations of $(1 : 120)^3$ (left) and $(1 : 120)$ (right). We observe that the persistent homology varies considerably between outputs of the constant probability model constructed with differing parameters, suggesting that the edge density of the network plays a large role in the persistent homology of this model.

To develop a deeper understanding of the generative and derived null models, we also calculated three common graph statistics on each model. Specifically, we compute the average degree, average clustering coefficient,[70] and modularity quality index[75] on each produced binary graph (that is, the binary graph at the end of the filtration). We show the results in Supplementary Fig. 6. Note that the binary graph at the final step of the filtration for the randomized nodes, decreasing degree, and distance from v_0 models is the same as that of the growing semantic feature network, so these are not separately included in our calculations.

Supplementary Notes

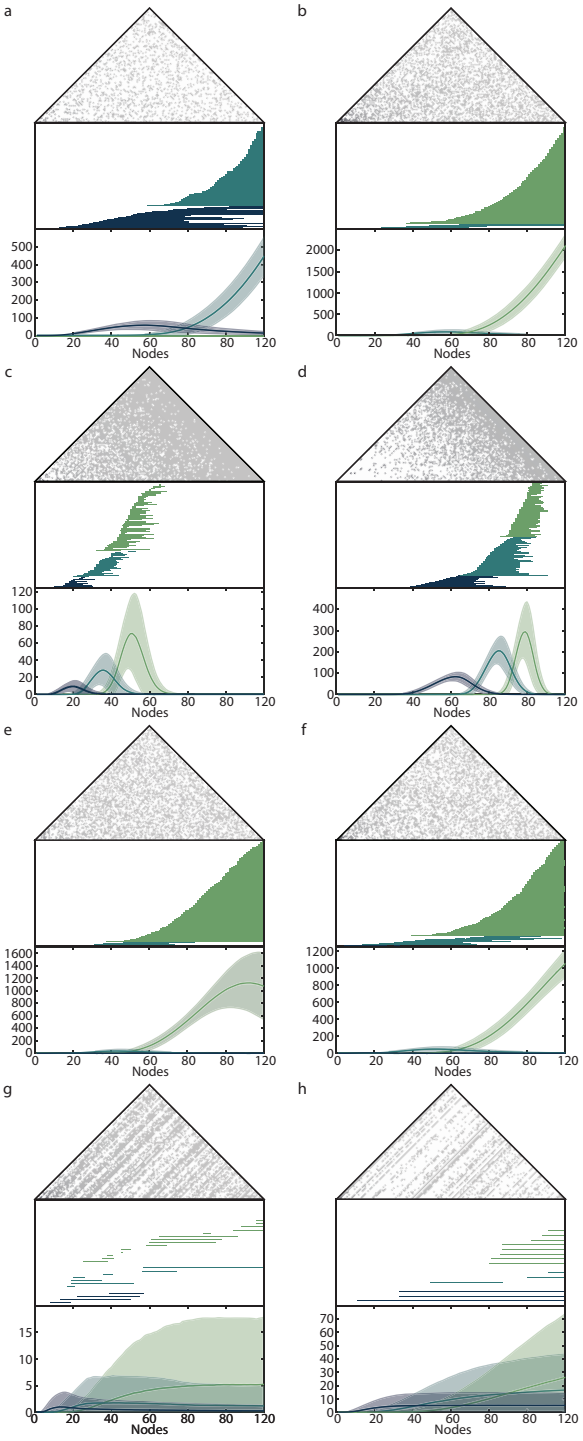
Details on grouped data

The results we present in the main text were constructed from grouped data following the methods described in.[4] One might ask the extent to which we can understand individual learning from our procedure. The data used in this paper gives one time-point per child, at which all words the child is able to produce are recorded. We hope in the future to perform our analysis on the growing semantic feature network of individual children. However, with the current data we do not yet have this resolution. Still, we can move towards predicting learning in an individual child by treating the current data as an empirical cumulative distribution and taking properties from the derived probability distribution. To this end, for each word we sample 100 individuals from a given month and record the number that can produce the word. We repeat this procedure 500 times for each month to construct a distribution of cumulative distribution functions (Supplementary Fig. 7, left). Using the average number of children producing the word for each month, we can then calculate the probability distribution and the mean. This mean can now be interpreted as the average month at which a child learns the word. Ordering the words by increasing expected month learned we obtain the persistence diagram and barcode in Supplementary Fig. 7 (right) and see that it very closely matches the results from the original ordering (Fig. 3). We expect the ordering here to better reflect the learning of an individual child, so this supports our hypothesis that cavities will form and close throughout learning.

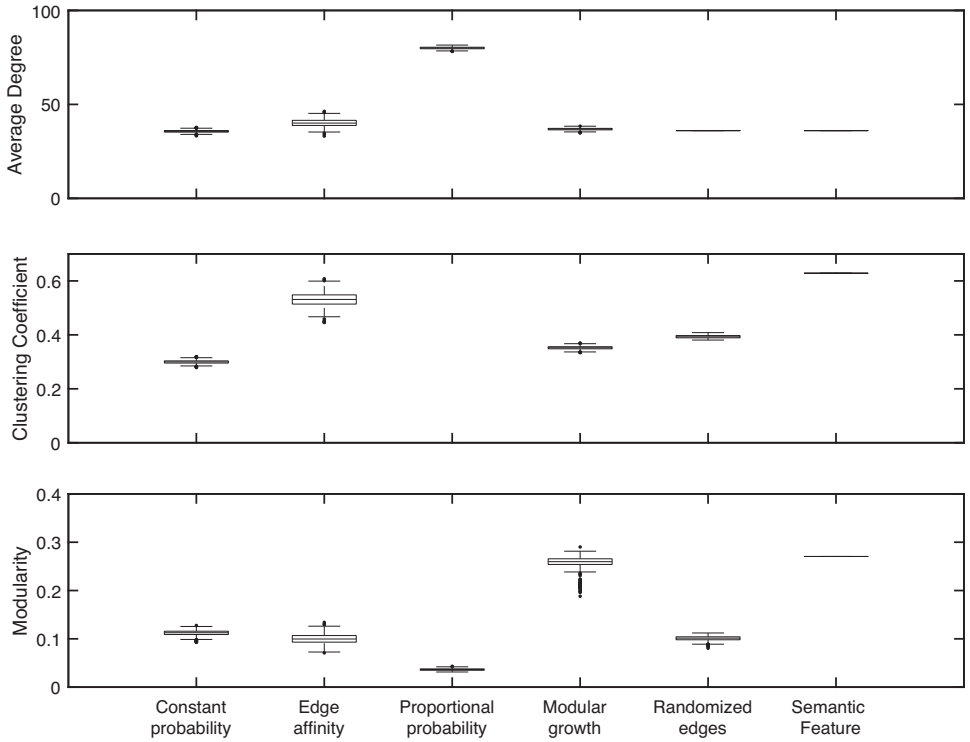
Further information for maternal education levels

First we address the distance between orderings. We calculate the maximum distance that any node moves between two orderings (also known as the L_∞ distance between orderings). As shown below in Supplementary Fig. 88, the distance between each pair is at least 35. We know by the Stability Theorem[76] that this is an upper bound for how far any point can move when comparing the persistence diagrams. Since the resulting barcodes are very similar (Supplementary Fig. 8), we can conclude that the persistent homology has changed little with respect to the worst possible scenario given by the maximum swap distance.

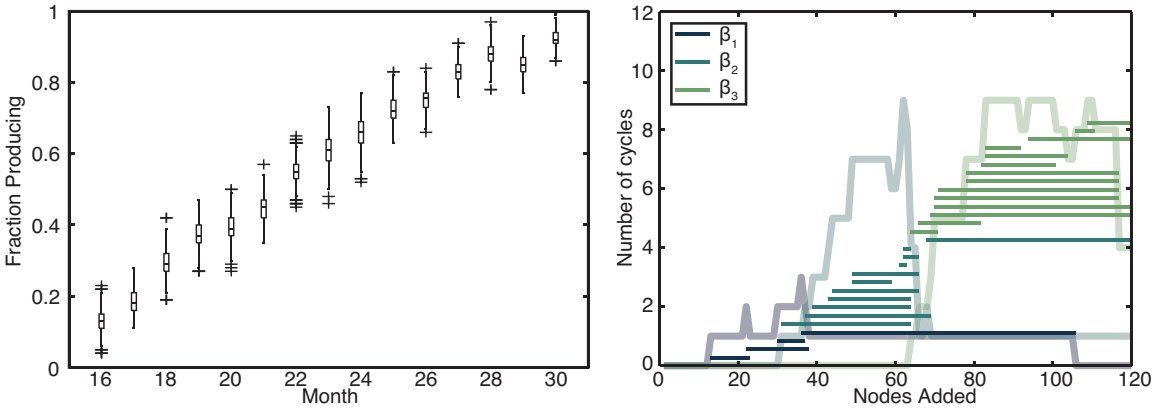
Additionally to supplement Fig. 4, we include in Supplementary Fig. 9 the barcodes of the *secondary*, *college*, and *graduate* growing semantic feature networks with the starting and ending words of each persistent cavity.



Supplementary Figure 5: **Persistent homology of model n-order complexes with varying parameters.** Results for the constant probability model with *(a)* $p = 0.2$ and *(b)* $p = 0.4$. Persistent homology for the proportional probability model with *(c)* $d = 0.5$ and *(d)* $d = 2$. Results for the modular model with *(e)* $p_{in} = 0.6$, $p_{out} = 0.4$ and *(f)* $p_{in} = 0.8$, $p_{out} = 0.2$. Persistent homology of the edge affinity model with affinity vector a random permutation of *(g)* $(1 : 120)$ and *(h)* $(1 : 120)^3$.



Supplementary Figure 6: **Comparison of common graph statistics between empirical data and both generative and derived null models.** Average degree (top), average clustering coefficient (middle), and modularity quality index (bottom) of generative models (left), derived models (middle), and the empirical data (right) studied in this paper. Central line marks the median while the top and bottom edges indicate the 25th and 75th percentiles, respectively.



Supplementary Figure 7: **Persistent homology using the ordering from repeated uniform subsampling of the original data.** *(Left)* Fraction of sampled children producing the word "box" at each month, created using 100 children per month and repeated for 500 trials. Central line marks the median while the top and bottom edges indicate the 25th and 75th percentiles, respectively. *(Right)* Barcodes and Betti curves generated from the persistent homology of the growing semantic feature network using the ordering on nodes created from calculating the expected month at which each word was learned.

Additional number of persistent cycles killed correlates

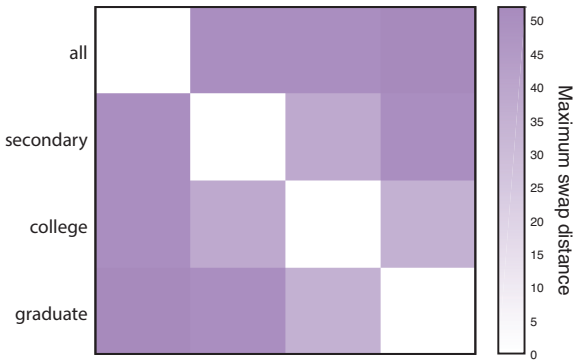
Frequency of a word in child-directed speech is known to correlate with age of acquisition.[27] We asked if caregiver output frequency correlates with the number of persistent cycles each node kills. We extracted frequency counts of child-directed speech from.[27,77,78] Of our original 120 words, 87 were found in this database so we restrict the following calculations to those 87 words. As described in the main text, we calculated the Spearman correlation coefficient between the persistent cycle death count and the node degree, clustering coefficient, betweenness centrality, word length, and word frequency for the semantic feature network using all children, and separately when broken down by education level (Supplementary Fig. 10). We observe a slightly decreasing trend of word frequency as corresponding nodes kill more persistent cycles, but this is not significant (Spearman correlation coefficient $df = 85$; *all:* $r = -0.2168$, $p = 0.0437$; *secondary:* $r = -0.2012$, $p = 0.0616$; *college:* $r = -0.1861$, $p = 0.0843$; *graduate:* $r = -0.0021$, $p = 0.9849$). Trends for number of persistent cavities killed with node degree, clustering, and betweenness are similar to those seen with all nodes (Fig. 5) in the growing semantic feature networks. To summarize, we observe the connectivity patterns of words better determine the tendency of a word to fill in a knowledge gap than do simple lexical features.

Additionally, prior work reports that the *relative feature distinctiveness* of a word is negatively correlated with the age at which the word is acquired.[17] The relative feature distinctiveness of a word is defined as

$$d(n) = \sum_{i=1}^{m} \frac{1}{w_i} \ ,$$

where node n has m features and where w_i is the number of words that have feature i. Intuitively, we can interpret the relative feature distinctiveness as a measurement of the rarity of a word's features within the given dataset.

One might hypothesize that learning words in the order of decreasing distinctiveness might produce more persistent cavities compared to the regular ordering, largely because we imagine that distinctive words might sit far from each other in the semantic feature network. Interestingly, we find that ordering nodes by decreasing distinctiveness produces Betti curves with quite similar peak magnitudes and peak locations to those observed in the original ordering (Supplementary Fig. 11, left). This observation stands in contrast to



Supplementary Figure 8: **Differences in word order between maternal education levels.** Maternal education levels shown on both axes. Each element in the matrix is the maximum swap distance of nodes between orderings derived from two maternal education levels. Colorbar indicates the magnitude of the maximum swap.

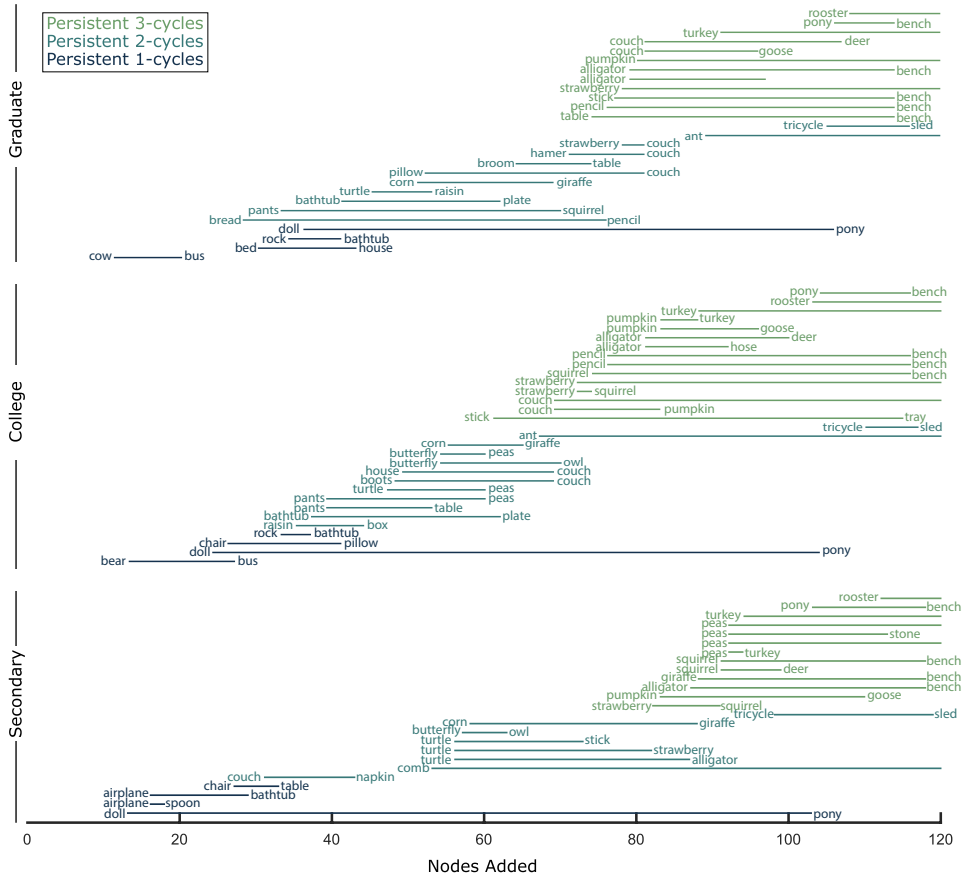
our original hypothesis about distinctiveness, but supports the notion that many word orderings will produce a similar progression of persistent features.

If a node has low relative distinctiveness, it shares many features with other nodes and thus is likely well-connected, making these low distinctiveness nodes candidates for persistent cycle killing. However, we find that the correlation between distinctiveness and number of cycles killed is not significant (Supplementary Fig. 11a, right; Spearman correlation coefficient $df = 118$; *all:* $r = -0.1910$, $p = 0.0366$; *secondary:* $r = -0.2183$, $p = 0.0166$; *college:* $r = -0.1993$, $p = 0.0291$; *graduate:* $r = -0.0835$, $p = 0.3643$), suggesting that words with low distinctiveness are not generally delayed in the learning if their neighbors have already been learned. It is interesting that we recover similar barcodes despite an L_∞ distance of 114 between orderings since swapping a node that far in the ordering could possibly prevent all observed persistent cycles from forming.

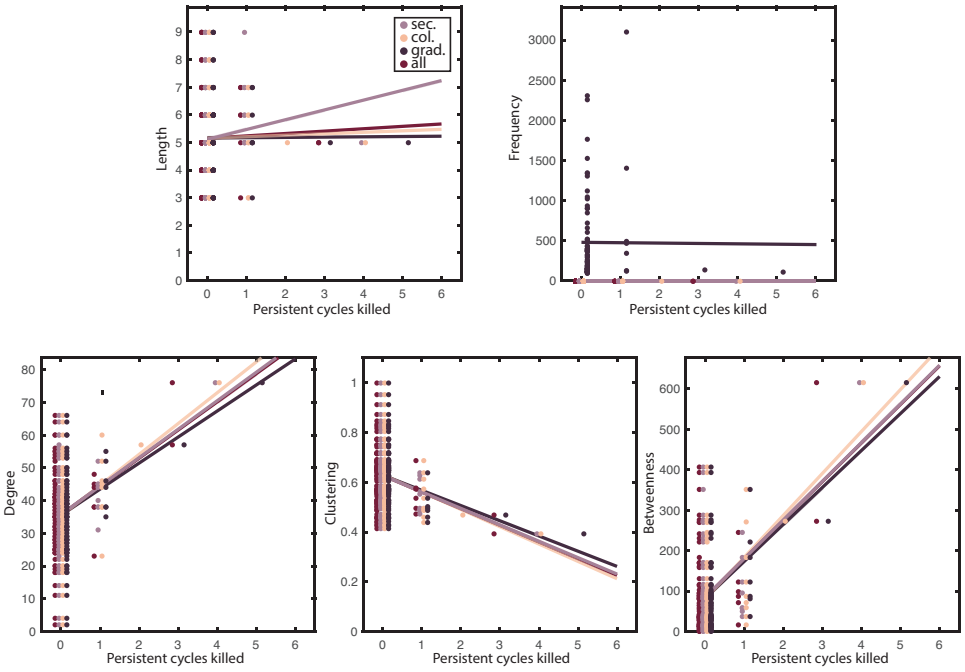
Finally, we also ask whether we observe persistent cavities under the modeling assumption that words are learned via the mechanism of preferential acquisition. Following,[18] we calculate the probability that each node is learned as

$$P(n) = \frac{(k_n + 1)^\beta}{\sum_i (k_i + 1)^\beta}$$

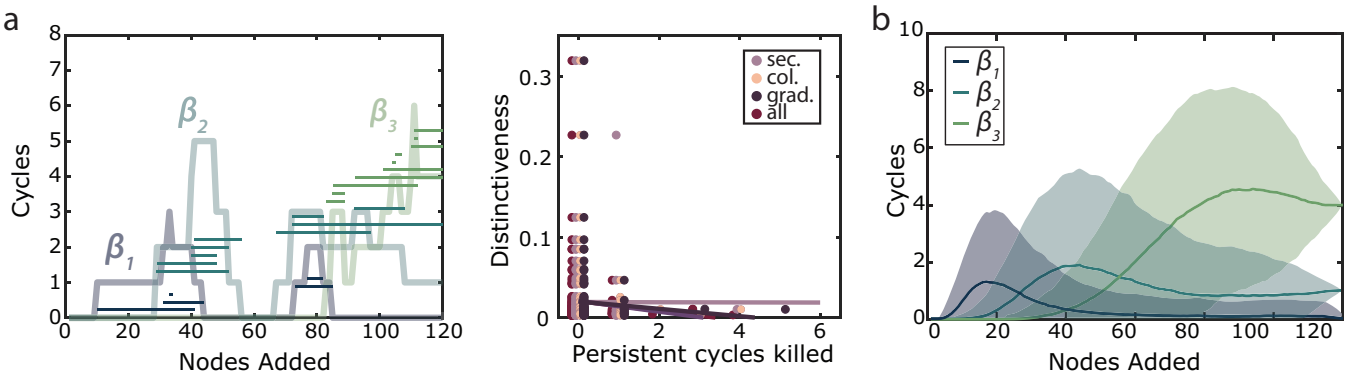
with $\beta = 1$. Words are sampled without replacement following this probability distribution. We repeat this process 1000 times and we show the resulting Betti curves in Supplementary Fig. 11b. We observe that these Betti curves are similar in peak magnitude and peak location to those observed in the semantic feature network. Together these findings support the claim that knowledge gaps are a robust feature of the learning process. Still, the relationship between knowledge gaps, word properties, and network properties is quite complicated and we suggest that further analyses are needed to tease apart potentially subtle but essential interactions. One concrete example would include weighting edges based on distances.[17]



Supplementary Figure 9: **Barcodes for *secondary*, *college*, and *graduate* growing networks.** Words corresponding to persistent cycle birth or death nodes depicted next to the associated bar.



Supplementary Figure 10: **Additional correlates of number of persistent cycles killed.** Number of persistent cycles killed by each node against corresponding word length, frequency in child-directed speech, node degree, clustering coefficient, and betweenness centrality.



Supplementary Figure 11: **Persistent homology when ordered by distinctiveness and preferential acquisition.** *(a) (Left)* Betti curves and barcodes from the persistent homology of the semantic feature network with nodes ordered by decreasing distinctiveness. *(Right)* Scatter plot of persistent cycles killed against the corresponding word distinctiveness. *(b)* Betti curves using preferential acquisition ordering in which words are more likely to be added if they have a high degree in the semantic feature network.