

PAPER • OPEN ACCESS

Encapsulation structure and dynamics in hypergraphs

To cite this article: Timothy LaRock and Renaud Lambiotte 2023 *J. Phys. Complex.* **4** 045007

View the [article online](#) for updates and enhancements.

You may also like

- [The distribution of the number of node neighbors in random hypergraphs](#)
Eduardo López
- [Dynamical systems on hypergraphs](#)
Timoteo Carletti, Duccio Fanelli and Sara Nicoletti
- [Higher-order co-mutation interactions in mitochondrial genomes](#)
Rahul K Verma, Pramod Shinde, Ankit Mishra et al.



PAPER

OPEN ACCESS

RECEIVED
31 July 2023

REVISED
16 October 2023

ACCEPTED FOR PUBLICATION
9 November 2023



PUBLISHED
22 November 2023

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Encapsulation structure and dynamics in hypergraphs

Timothy LaRock^{1,*}  and Renaud Lambiotte^{1,2} ¹ Mathematical Institute, University of Oxford, Oxford, United Kingdom² Turing Institute, London, United Kingdom

* Author to whom any correspondence should be addressed.

E-mail: larock@maths.ox.ac.uk**Keywords:** higher-order networks, hypergraphs, line graphs

Abstract

Hypergraphs have emerged as a powerful modeling framework to represent systems with multiway interactions, that is systems where interactions may involve an arbitrary number of agents. Here we explore the properties of real-world hypergraphs, focusing on the encapsulation of their hyperedges, which is the extent that smaller hyperedges are subsets of larger hyperedges. Building on the concept of line graphs, our measures quantify the relations existing between hyperedges of different sizes and, as a byproduct, the compatibility of the data with a simplicial complex representation—whose encapsulation would be maximum. We then turn to the impact of the observed structural patterns on diffusive dynamics, focusing on a variant of threshold models, called encapsulation dynamics, and demonstrate that non-random patterns can accelerate the spreading in the system.

1. Introduction

Networks provide a powerful language to model and analyze interconnected systems [48]. The building blocks of networks are pairwise edges, and these blocks can then be combined to form walks and paths, making it possible for systems to be globally connected yet sparse. Since the seminal work of Watts and Strogatz 25 years ago [64], a key focus of network science has been to investigate the relationship between the structure of a network and the dynamics taking place on its nodes [30]. This program requires the design of metrics to capture significant, non-random structural properties of networks, e.g. the clustering coefficient, the degree distribution or modularity, as well as the specification of dynamical models, both linear and non-linear, for the diffusion between neighbouring nodes. An important observation is that the same structural property may affect different dynamical models in different ways, e.g. a high density of triangles tends to slow down simple diffusion, but facilitate complex diffusion [20].

Finding the right modeling framework for interacting systems is a challenging task. While networks have the advantage of simplicity, it has been recognized that they may also neglect critical aspects of a system and even lead to a misleading representation. Driven by the availability of datasets with richer connectivity information in recent years, different frameworks have emerged to enrich the network representation, leading to different types of higher-order networks [29]. One branch of this research has extended pairwise graph-based models to multiway interaction frameworks, most notably as hypergraphs or simplicial complexes, to account for group interactions among arbitrary numbers of nodes [3, 7, 53].

Multiway interactions naturally appear in many systems, ranging from social interactions, where people interact in groups rather than in pairs [50], to joint neuronal activity in brains [52] and cellular networks [28]. Different computational tools have been adapted to multiway systems, for instance for centrality measures [4] and community detection [65]. Researchers have also investigated how the structure of multiway interactions impacts dynamical processes [6, 8, 12, 26, 31, 34, 43, 55], especially the conditions under which dynamics on hypergraphs and simplicial complexes differ from those on networks [10, 46, 47].

The objectives of this paper are twofold: to propose metrics that characterize the non-random patterns of *encapsulation* in multiway systems, and to explore dynamical models that may be affected positively or negatively by this type of hypergraph structure. These objectives are motivated by a well-known conceptual

difference between the two main representations for multiway systems, hypergraphs and simplicial complexes. By definition, a simplicial complex of size k nodes includes all of the subfaces of the complex. In contrast, a hyperedge of size k nodes does not imply the existence of any subsets as hyperedges in the same hypergraph. We refer to this difference as the *simplex assumption*. For example, using a simplicial complex to represent the relationship between three nodes $\{a, b, c\}$ assumes that the subfaces $\{a, b\}$, $\{a, c\}$, $\{b, c\}$ all exist, along with the individual nodes. This is a strong assumption that is unlikely to hold, even approximately, in real data. A classic example is co-authorship, where a jointly authored paper between three co-authors does not imply that each pair of co-authors have also authored separate papers together, nor that each co-author has published a single-author paper. Recent work has investigated the relationship between these two representations [2], and shown that the choice of higher-order representation does effect the outcome of dynamical processes [51, 60, 66].

Simplicial complexes and hypergraphs can be seen as poles on a spectrum of multiway interaction structure, and it is likely that real data falls somewhere in-between. In this work, we build on previous investigations of this spectrum of overlapping higher-order structures, as well as random models for hypergraphs and simplicial complexes [9, 11, 14, 15, 17, 27, 37, 60]. Our approach begins with the construction of a *line graph*, where nodes are the edges of the original (hyper)graph and there is a link between two nodes if their corresponding (hyper)edges have a node in common [1, 19]. The interactions between hyperedges of arbitrary sizes make it possible to define a variety of different line graphs for hypergraphs. As each hyperedge can be seen as a set of nodes, choosing a relation to determine the edges of the line graph is equivalent to choosing how to compare sets, leading to multiple definitions of line graphs for hypergraphs [1, 13, 41].

We will focus in particular on what we refer to as an *encapsulation graph*, where two hyperedges are connected (by a directed edge from larger to smaller) if one is the subset of the other. We then analyze the properties of the resulting directed acyclic graphs (DAG) built from real-world hypergraphs and from a synthetic hypergraph model called the Random Nested Hypergraph Model (RNHM) [27], which allows for some control over the extent of nested structure through random rewiring of simplicial complexes. Finally, we define a process for the spread of a complex contagion on a hypergraph through its hyperedges, and show how varying levels of encapsulation structure impact the spread of the contagion in both synthetic and real hypergraphs.

2. Measuring overlap and encapsulation in hypergraphs

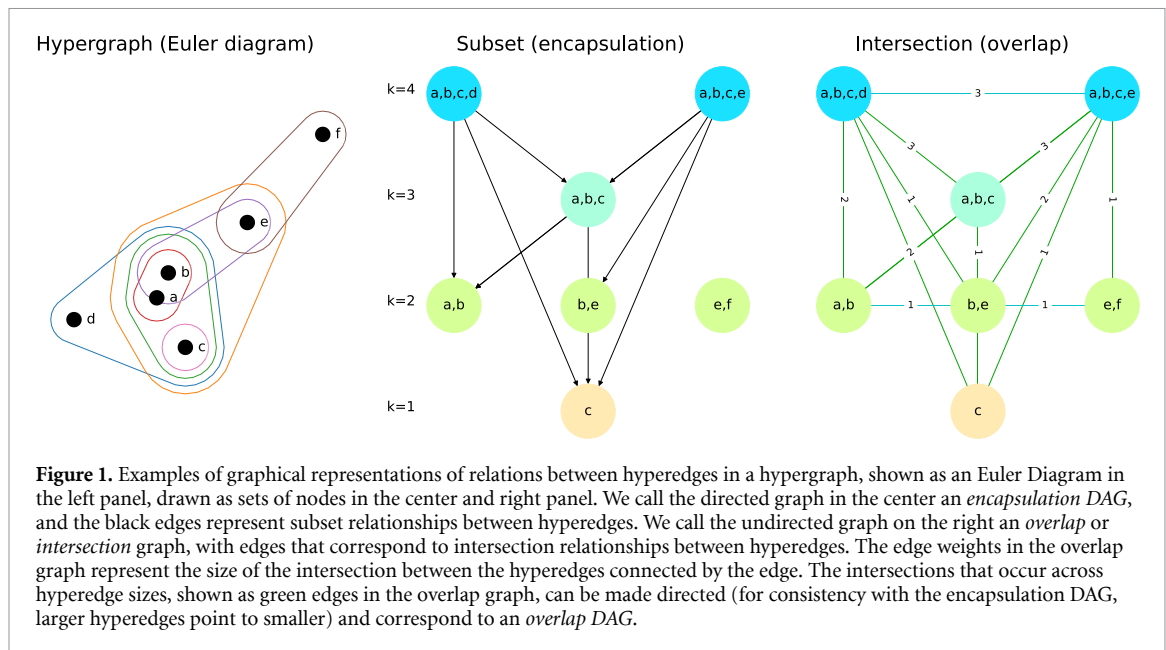
Consider a list of multiway interactions, where each item in the list is a set of nodes that represent a group interaction. We will represent these interactions as a hypergraph, and focus in particular on *aggregated, static* hypergraphs, where all interactions are included regardless of any dynamic or temporal information. In fact, for all of the empirical datasets we will examine, this static hypergraph is actually the result of aggregating interactions that happen over time. We will also make our hypergraphs *simple*, meaning that no edges are repeated, i.e. hyperedges are contained in a set, rather than a multiset. In the future, the techniques we develop here could be extended to study the relationships between hyperedges over time, extending, for example, work on simplicial closure [5] or temporal dynamics of group interactions [25, 57], or to weighted multiway interactions [56, 57].

Formally we represent the multiway interactions as a hypergraph $H = (V, E)$ where $V = 1, 2, \dots, n$ is the set of n nodes and $E = \{e_1, e_2, \dots, e_m\}$ is the set of m hyperedges representing interactions between the nodes in V , with the size of each interaction measured as the number of nodes and represented by $s_i = |e_i|$.

To understand the extent of nestedness in the structure of a hypergraph, we build a line graph where the nodes are hyperedges and where there is a directed link between two hyperedges if one is a subset of the other. These links represent what we call *encapsulation* relationships between hyperedges. More formally, given two hyperedges e_i and e_j such that $s_i > s_j$, we say e_j is encapsulated by e_i if $e_j \subset e_i$.

The line graph representing encapsulation relationships is a DAG where a directed edge between two hyperedges means that the larger encapsulates the smaller. Since for every connected e_i and e_j we know that $s_i > s_j$, a cycle in this graph would imply that a smaller hyperedge encapsulates a larger hyperedge, which is impossible, thus the graph is always a DAG. We refer to this DAG as the *encapsulation DAG* of a hypergraph and the center panel of figure 1 shows an example. The number of edges in the encapsulation DAG is the number of encapsulation relationships present in the hypergraph. By construction, a hypergraph corresponding to a simplicial complex would have the maximum possible number of edges given the facet sizes.

The encapsulation DAG is closely related to a Hasse Diagram representing a partial ordering of a set of sets. However, a Hasse Diagram is transitively reduced by construction, meaning that an edge between two



nodes is removed if there is an alternative path between the nodes. Hasse Diagrams with weights associated to their nodes have been used to define weighted simplicial complexes of hypergraphs, which were further used to predict the evolution and recurrence of small groups [56, 57]. While we will examine transitively reduced encapsulation DAGs in section 3.2, for consistency we will refer to the line graph with edges representing encapsulation relationships as an encapsulation DAG throughout.

The encapsulation DAG only one way to study encapsulation relationships. In [40], similar statistics to those we propose below were measured without explicitly constructing the DAG object, finding that biological and co-authorship hypergraphs do not have more nestedness relationships as hyperedge sizes increase, while social and technological networks have more nesting relationships as hyperedge size grows. We will confirm some of these findings in section 3.1. Similarly, the concept of *simpliciality* of multiway interaction data, which is closely related to the encapsulation DAG and measures derived from it that we propose below, has recently been introduced independently in [35].

As mentioned previously, other line graphs can be defined by considering different relations between hyperedges. An important relation is the *intersection* between the hyperedges, which defines an *overlap graph*. Given two hyperedges e_i and e_j , an undirected edge exists between them if $|e_i \cap e_j| > 0$, and the weight of the edge is the size of the overlap $|e_i \cap e_j|$ (or, alternatively, normalized as $\frac{|e_i \cap e_j|}{\min s_i, s_j}$). If we remove the edges between hyperedges of the same size and impose directionality on the remaining undirected edges, for example by directing edges from larger hyperedges to smaller, we obtain a DAG that we call an *overlap DAG*. The right graph in figure 1 shows an example of the intersection relation. An overlap graph with edge weights corresponding to the Jaccard distance between hyperedges has been used to uncover link communities using hierarchical clustering [41]. We note that overlap graphs are also related to clique-graph representations of pairwise networks [18].

In appendix A we give detailed descriptions including pseudocode for the algorithms we use to construct overlap and encapsulation structures. We show that the worst-case time complexity is bounded by the product of the number of hyperedges m , the largest hyperedge size s_{\max} , and the largest node degree k_{\max} . We also discuss and informally evaluate an alternative strategy using the set-trie data structure [54].

Let us make a short digression about dynamics here, a topic that we will cover in more detail in section 4. The *encapsulation* and *intersection* relations capture different ways in which hyperedges may be related with each other, but they also have different implications for dynamical processes on the hypergraph. The *intersection* graph is compatible with dynamics centered on the nodes of the hypergraph. One can think here, for instance, of a threshold model where all of the nodes in a hyperedge become activated if a certain number (or fraction) of its nodes are already activated. The intersection graph then provides us with information on how the activation of one edge may spread into others. On the other hand, the encapsulation DAG is more naturally associated to dynamics where the states are defined on the hyperedges and influence spreads explicitly through interactions (as opposed to nodes), in a way reminiscent of the Hodge Laplacian for diffusive processes [55]. A more thorough discussion of the interpretation of this type of dynamics, and its simulation on both synthetic and real-world hypergraphs, will be given in section 4.

Table 1. Number of nodes, number of hyperedges, the density in the projected graph, and number of DAG edges in the largest connected component of each empirical hypergraph after removing multiedges. The same statistics for the full datasets are available in appendix B.

Dataset	n_{cc}	m_{cc}	Proj. density	DAG edges
coauth-MAG-Geology	898 648	947 977	10^{-5}	1650 117
coauth-MAG-History	219 435	205 531	10^{-5}	217 627
contact-high-school	327	7818	0.11	7942
contact-primary-school	242	12 704	0.29	16 199
email-Enron	143	1512	0.18	8240
email-Eu	979	25 008	0.06	277 224

3. Encapsulation in empirical data

In this section we introduce basic measurements of encapsulation relationships in some empirical hypergraph datasets, all of which were made available online with the publication of [5]. We focus in particular on coauthorship [58], social contact [44, 59], and email communication datasets [38, 65]. In table 1, we show some statistics of the largest connected components of the hypergraphs. Following [5], we exclude hyperedges of size greater than 25 nodes to keep some amount of consistency across the datasets. As mentioned above, we also ignore multiedges in the datasets and therefore consider the simple hypergraph representation of each.

The coauthorship datasets, which include decades of published papers in multiple fields, contain numbers of nodes and edges that are multiple orders of magnitude larger than the face-to-face contact and email datasets. They are also orders of magnitude less dense in terms of the proportion of edges that exist in the projected graph where an edge exist between two nodes if they occur in the same hyperedge at least once.

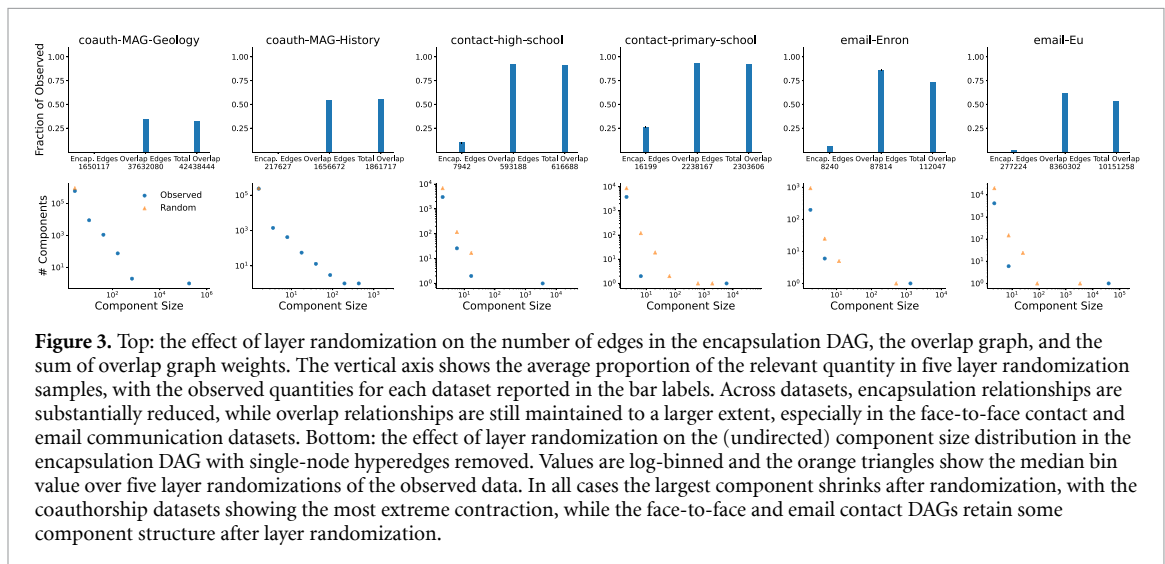
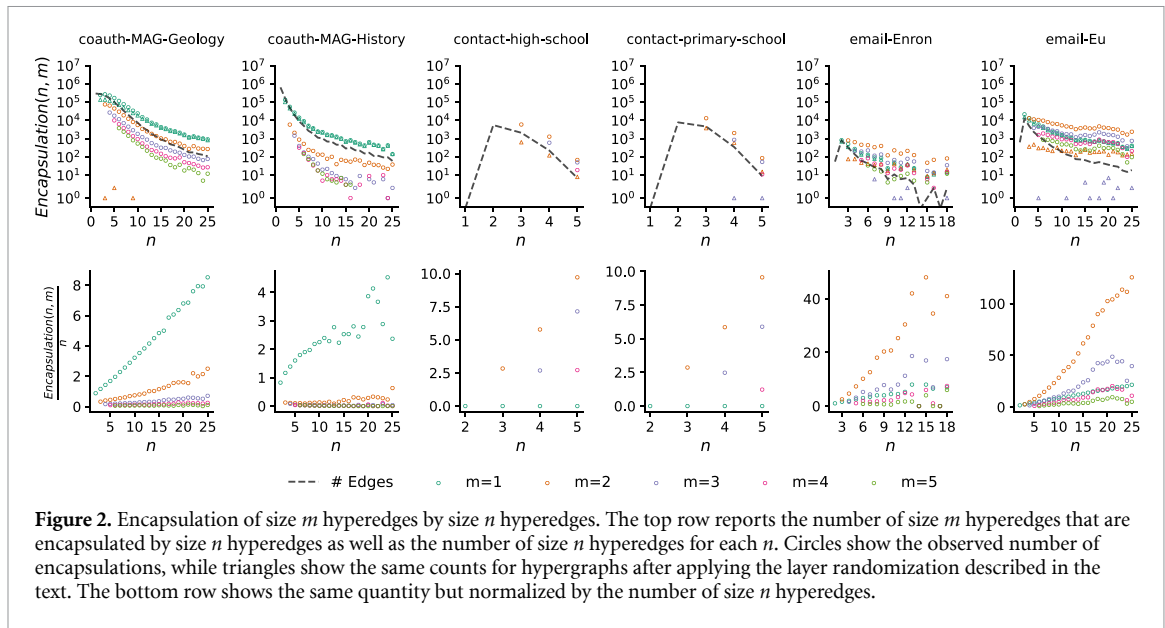
3.1. Degree in the encapsulation DAG

For each hyperedge, we are interested in the extent to which it encapsulates other hyperedges present in E , or equivalently we are interested in its out-degree in the encapsulation DAG. In the top row of figure 2, we report the total number of hyperedges of each size m that are encapsulated by hyperedges of larger sizes $n > m$. The total number of hyperedges of each size n is shown as a dotted line.

For each m , the number of observed hyperedges encapsulated decreases with n , but so does the number of size- n hyperedges. To account for the distribution of hyperedge sizes, in the bottom row of figure 2 we report the same counts but divided by the number of size- n hyperedges, giving us the number of encapsulated size- m hyperedges per size- n hyperedge.

We also show the same quantity in a randomization of the hypergraph which we call the ‘layer randomization’. The name comes from the fact that in this randomization procedure we view the sets of hyperedges of each size k as a *layer*, similar to the multiplex approach taken in [60], the uniform filtering method described in [32], and the benchmark randomization method proposed in [45]. The procedure then works as follows: for each layer of the hypergraph consisting of hyperedges of size k , we gather all of the hyperedges and the set of their constituent nodes, then shuffle the labels of the nodes. We repeat this procedure for every layer independently. The result is a hypergraph where the hyperedge size distribution and the unlabeled node degree distribution within each size-layer are preserved, but the labeled node degree distributions within size-layers, the node hyperdegree distribution, and, most importantly, the cross-size encapsulation and overlap relationships are randomized. In other words, we randomize the hypergraph across layers, but not inside layers. This is the reason why we opted for this randomization procedure, and not, for example, the configuration model for hypergraphs introduced by [11]. Future work could investigate the effect of other randomization procedures such as those discussed in [11, 60]. In figure 3, we show the proportion of encapsulation and overlap relationships destroyed by the layer randomization as well as the change in component size distribution of the encapsulation DAG.

Only the coauthorship datasets include hyperedges of size 1 (i.e., nodes representing papers authored by a single individual). The number of encapsulations of 1-node hyperedges increases with n after accounting for the number of size- n hyperedges across all three datasets. This indicates that authors who are part of large collaborations also publish single author papers. However, the relationship is not as strong as would be expected under the simplex assumption. In that case, every node should appear as a zero-simplex and the number of encapsulations would grow exactly as $y = n$ since all n nodes would be encapsulated for every size- n hyperedge. Instead the encapsulation relationship for single nodes grows sublinearly, indicating that there are many nodes which appear in hyperedges of size larger than 1, but never appear alone. Note also that the layer randomization does not substantially reduce the number of encapsulations of one-node hyperedges, since the shuffling of the one-node layer has no effect, and shuffling at each higher layer still

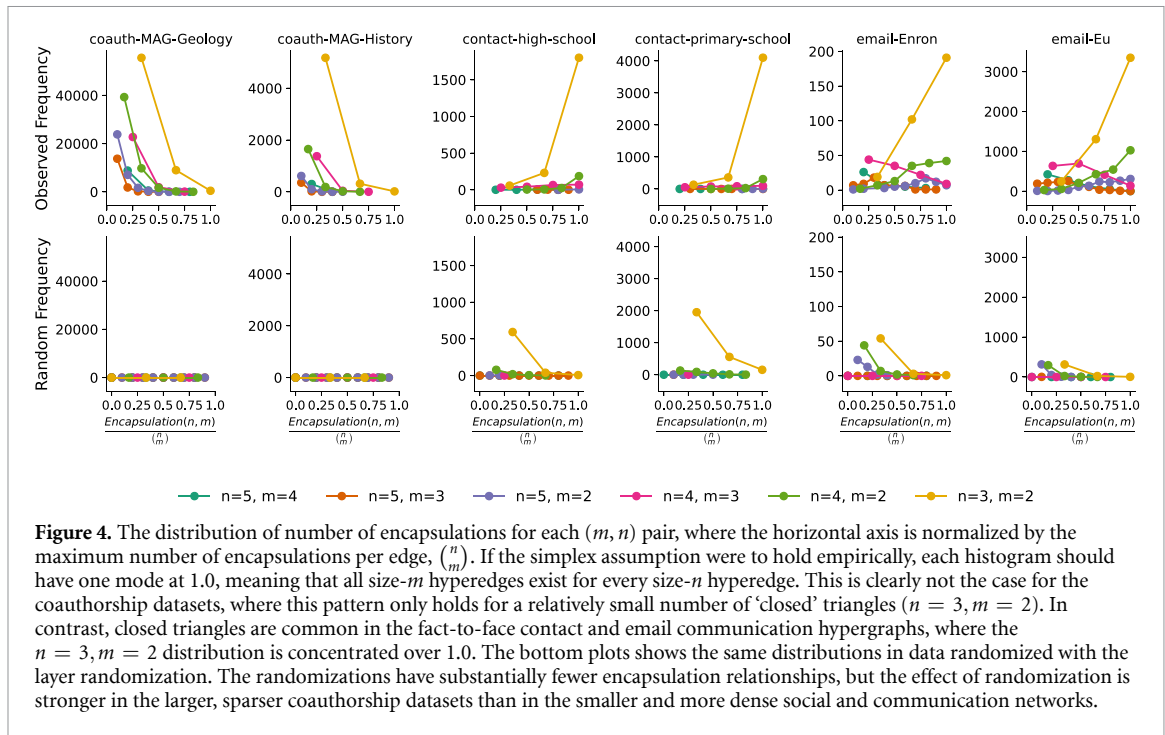


results in some encapsulations of one-node hyperedges necessarily, since the set of nodes in each layer does not change, only which positions the nodes occupy in the structure.

The relationship is even weaker for larger values of m . In this case, the simplex assumption would lead to the relationship $y = \binom{n}{m}$, since for every size- n hyperedge all possible size- m hyperedges would have to exist. However, for all values, the number of encapsulations per size- n hyperedge stays well below 1, meaning that, on average, a size- n hyperedge encapsulates few smaller hyperedges relative to its maximum capacity. Notably, for all of the coauthorship datasets, encapsulation relationships tend to be destroyed among hyperedges of any size after the layer randomization is applied, as expected.

The encapsulation structure of the face-to-face social contact hypergraphs appears to be more sparse than the rest of the datasets, partly due to the fact that there are fewer large interactions with a maximum interaction size of only five nodes. However, even with this more sparse structure, there are substantial encapsulation relationships, especially for hyperedges with 2 and 3 nodes.

The email communication hypergraphs show a substantially nested structure where large group emails are composed of groups with many smaller interactions in separate email chains, especially pairwise and three-node interactions. This is consistent with an intuitive understanding of how email communication works within organisations: many small group email chains will naturally occur to facilitate day-to-day operations and side conversations, while large group emails will occur around big meetings, decisions, or announcements that involve larger proportions of the organisational structure. Interestingly, compared to the coauthorship data, the layer randomization keeps substantially more of the encapsulation relationships in



the email communications. We hypothesize that this is due to the smaller number of nodes in the email datasets, which constrains the possible randomizations.

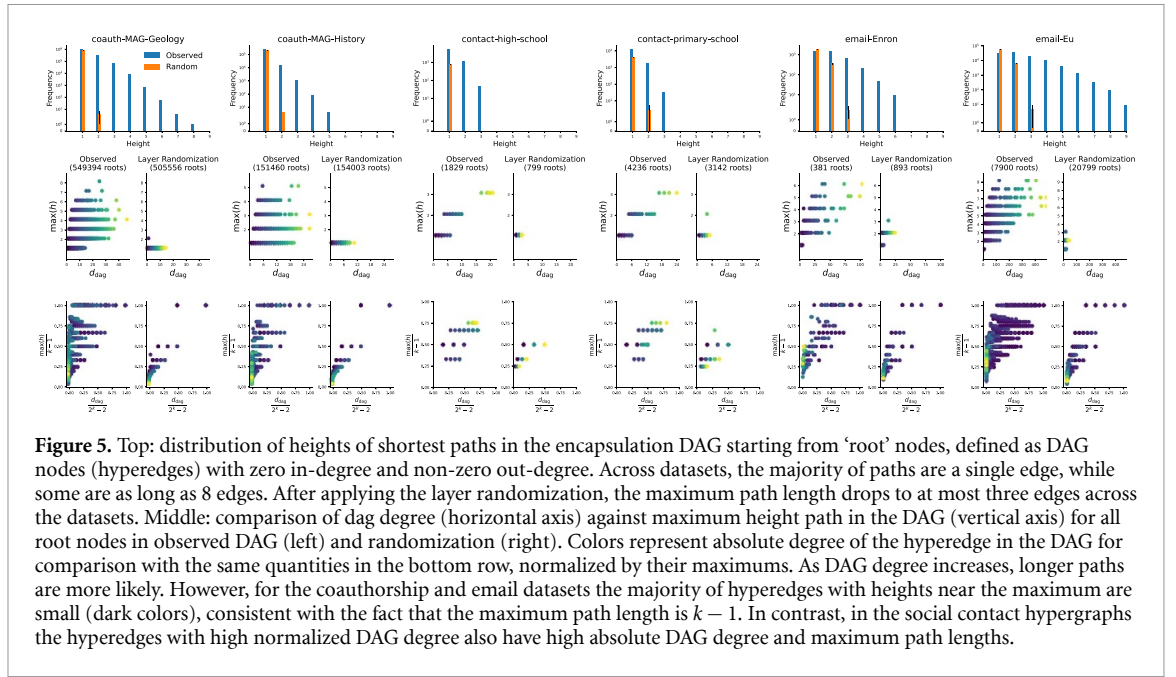
In figure 4, we show the distribution of encapsulation for each (n, m) pair; that is, one distribution for each point in figure 2 up to $n = 5$. We compute for each $\alpha \in E$ with $s_\alpha = n$ the number of out-neighbors of α in the encapsulation DAG that are of size m . We then normalize this quantity by the maximum number of subsets of size m , which is $\binom{n}{m}$. Thus if a histogram is fully concentrated on 1, there is full encapsulation and the simplex assumption holds. The bottom row of figure 4 shows the same histograms computed on the layer randomized version of the hypergraph. As we observed in figure 2, the number of encapsulations decreases for all of the coauthorship datasets when n increases. The distributions in figure 4 show that the most common amount of encapsulation is exactly one subset (leftmost point of each line), and relatively few hyperedges fully encapsulate all of the possible subsets (rightmost point in each line). However, we observe the opposite pattern in the social contact and email datasets, where full encapsulation of two-node hyperedges by three-node and four-node hyperedges is common in the observed data, and these relationships are destroyed by the layer randomization.

3.2. Paths through encapsulation DAGs

In this section we show how analysis of encapsulation DAGs can help understand the structure of encapsulation relationships. An encapsulation DAG encodes interaction structure in at least three ways. As shown above, we can use the out-degree of a hyperedge in the DAG to measure the extent to which subsets of that hyperedge also appear as hyperedges. Similarly, the in-degree of a hyperedge in the DAG indicates the extent to which the supersets of a hyperedge exist, e.g. how much a given hyperedge is encapsulated. Finally, and this is the purpose of this section, the length of paths in the DAG indicates the ‘depth’ of encapsulation relationships.

Here we analyze the height of *rooted paths in the transitively reduced DAG*, inspired by the approach taken in [61]. A rooted path is one that begins from a root node, which we define as a node in the DAG with zero in-degree and non-zero out-degree. We consider paths starting from root nodes because they indicate the maximum possible path lengths through the DAG. A transitively reduced DAG is one in which all edges representing shorter redundant paths are removed. For example, if we have the edges A-B, B-C, and A-C, in the transitively reduced DAG the edge A-C would be removed, since there would still be a path from A to C without that edge. Analyzing the DAG after removing these ‘shortcut’ edges gives us a sense for the extent to which intermediate sized hyperedges are or are not present.

The distribution of path lengths in the transitively reduced DAG indicates the depth of the encapsulation relationships in the hypergraph. If the distribution is skewed towards the maximum length ($k - 1$ edges for a



hyperedge on k nodes), this indicates a hierarchy of encapsulations in the sense that multiple intermediate hyperedges of different sizes are all encapsulated by the same larger hyperedge (the root). In contrast, if most path lengths are short, this indicates that encapsulation relationships in the hypergraph are concentrated between only two different sizes at a time, a kind of shallow encapsulation. Note that transitively reduced DAGs corresponding to two hypergraphs with very different encapsulation structures could have similar numbers of edges, but very different path length distributions. As we will discuss below, deeper and more hierarchical encapsulation relationships can have important implications for how a contagion can spread over the hyperedges of a hypergraph.

In the top row of figure 5, we show the distribution of heights in each dataset compared to the average over multiple layer randomizations. After randomization, the maximum path length through the transitively reduced DAG drops substantially in every dataset, and the number of paths of length two drops by multiple orders of magnitude in all of the coauthorship and contact datasets, but not in the email datasets.

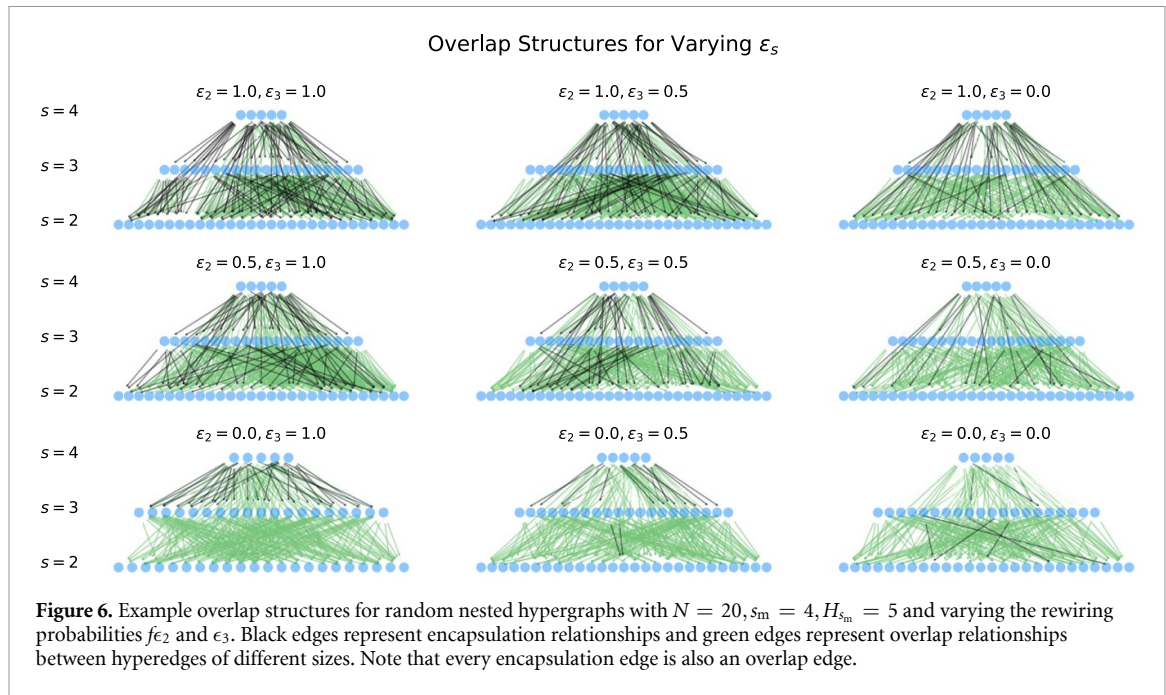
In the middle and bottom rows of figure 5, we plot for each root hyperedge its degree in the DAG against its maximum height (path length) in the transitively reduced DAG. The middle row shows the relationship without normalization for the observed (left) and layer randomized (right) hypergraph. The DAG degree of a hyperedge and its maximum length path in the transitively reduced DAG are positively correlated to varying extents across all of the datasets, but in the coauthorship datasets there are many hyperedges with high DAG degree that have relatively low maximum path lengths of only 2 or 3 edges.

As mentioned previously, the maximum height is bounded by $k - 1$, where k is the size of the hyperedge, since the maximum path length will pass through exactly one node (hyperedge) of each size $0 < k' < k$, of which there are $k - 1$. The bottom row of figure 5 again shows the relationship between DAG degree and maximum height, but with both quantities normalized by their maximums.

As expected, when a root hyperedge has maximum degree in the DAG, it also has maximum path length (the opposite need not hold). The dark colored points in the top right of each normalized scatter plot indicate that only the hyperedges with small degrees have the maximum degree, meaning that they are also small hyperedges.

3.3. RNHM

In this section we describe the RNHM developed in [27], which we will use as a starting point for analyzing the relationship between nested hypergraph structure and a hyperedge contagion process. The parameters of the model are: the number of nodes N ; the maximum sized hyperedge s_m ; the number of hyperedges of size s_m , denoted H_{s_m} ; and ϵ_s , the probability of rewiring a hyperedge of size $s < s_m$. Hypergraphs generated by this model are sampled by the following process. First, H_{s_m} hyperedges of the maximum size s_m are sampled, where the probability of a node being included in a hyperedge is uniform. Second, all of the subsets of those hyperedges (i.e. the powerset of every edge excluding sets with size less than 2) are added to the hypergraph.



In some simulations, we also include all of the individual nodes as one-node hyperedges. Finally, each of the encapsulated hyperedges with size $1 < s < s_m$ are rewired with probability $1 - \epsilon_s$, meaning that when ϵ_s is small, hyperedges of size s are more likely to be rewired.

Rewiring a hyperedge involves (i) choosing a pivot node in the edge uniformly at random; (ii) deleting all other nodes from the edge; and (iii) replacing the deleted nodes with nodes chosen uniformly at random from outside of the hyperedges that are supersets of the original edge, ensuring that the new edge does not already exist in the hypergraph. Since this model will be used as a substrate for contagion dynamics in the next section, we further constrain the RNHM by rejecting hypergraphs that are not connected.

In figure 6 we show DAG representations of random nested hypergraphs, where edges of the encapsulation DAG are drawn in black and edges from the overlap DAG are drawn in green. As ϵ_s decreases, so does the number of encapsulation relationships (DAG edges). When $\epsilon_s = 1$, no hyperedges are rewired, so all encapsulation relationships exist. As ϵ_s decreases, rewiring of hyperedges reduces the number of encapsulation relationships until, when $\epsilon_s = 0$, almost no encapsulation relationships between four-node and s -node hyperedges exist. However, since the s -node hyperedges were constructed based on the set of nodes that appeared in the four-node hyperedges, some encapsulation relationships may randomly remain after rewiring.

4. The role of encapsulation structure in dynamics

In this section we show that encapsulation plays a role in modulating the relationship between higher-order interactions and dynamical processes. We study a complex contagion process for which encapsulation and overlapping structures are vital to spreading. It is important to emphasise that our analysis focuses on a purely higher-order effect, as the notion of encapsulation has no counterpart in classical networks.

We study a hypergraph complex contagion process where in each discrete timestep, every node $u \in V$ and hyperedge $\alpha \in E$ in the hypergraph is in a binary state, either inactive or active. We represent these states using two binary vectors, s_u for nodes and x_α for edges, which both take a value of 0 if the corresponding node or edge is inactive, and 1 if active. At each time step, an inactive hyperedge α , $x_\alpha = 0$ is activated if more than a threshold τ of hyperedges which it directly encapsulates, i.e. hyperedges of size $|\alpha| - 1$, are also active. Therefore activation can only spread in hypergraphs with an encapsulation structure that is tightly nested, with many encapsulation relationships between adjacent layers of the DAG. We refer to this class of

contagion as *encapsulation dynamics* and focus on two variants depending on the influence we allow individual nodes to have on the dynamics³.

In the first variant, which we refer to as *strict* encapsulation dynamics, individual nodes can only have influence in the dynamics if they appear in the hypergraph as a 1-node hyperedge. These 1-node hyperedges only appear in the coauthorship datasets, meaning that in the other datasets individual nodes have no influence and node state has no bearing on the spreading process. In contrast, in the *non-strict* variant we allow any individual node to influence pairwise interactions in which it participates. This corresponds to an assumption that all individual nodes are also 1-node hyperedges in the hypergraph and makes the state of individual nodes relevant to how the process can evolve. It also allows for exactly one kind of ‘backwards’ activation, since activation of a large hyperedge will activate the individual nodes, while in general we do not allow activation of a large hyperedge to activate any of its subhyperedges in encapsulation dynamics. Instead, all activation flows upward through the encapsulation DAG from smaller to larger hyperedges. For a further discussion of the possible variants of encapsulation dynamics, see appendix C.

Intuitively, encapsulation relationships are necessary to the spreading process in these dynamics, since larger hyperedges can only be activated if they encapsulate smaller hyperedges, which in turn must encapsulate still smaller hyperedges. We make an analogy between this process and building a campfire, where the smallest hyperedges correspond to dry leaves and twigs, medium hyperedges correspond to kindling, and the largest hyperedges correspond to the logs. Thus the ‘goal’ of the encapsulation dynamical process we have defined is to catch the logs on fire by first lighting the fuel.

The encapsulation dynamics can be seen as a generalisation of threshold models, which have been studied systematically in the context of opinion dynamics on graphs [63] and hypergraphs [16, 23, 26, 49]. An important difference is that only activated nodes that are all connected by an active hyperedge can activate a larger hyperedge. From an opinion dynamics perspective this could be interpreted as follows: a set of nodes that is part of a larger set may change their collective behavior only if nodes in the smaller set form an interacting unit, which allows them to coordinate their action. This moves the threshold dynamics from the level of nodes, where influence depends on how many neighbors of a node are active, to the level of hyperedges, where influence depends on how many subinteractions of the interaction are active. Take for example the hyperedge $\{e, f\}$ in figure 1. In a traditional threshold model, activating the nodes in $\{e, f\}$ may result in activation of the hyperedges $\{a, b, c, e\}$ and $\{b, e\}$, and trigger a cascade of activations in the hypergraph. Similarly, in the non-strict version of encapsulation dynamics, the activation of e could activate $\{b, e\}$ and lead to full activation. The potential influence of $\{e, f\}$ on other hyperedges can be gleaned from its connections in the interaction graph (right of figure 1). However, the picture is strikingly different in the encapsulation case, where $\{e, f\}$ is disconnected in the DAG and therefore has no impact on future activations. Indeed, from that perspective, it is not the fact that node e is activated that matters, but instead that smaller hyperedges encapsulated in larger hyperedges are activated. Thus we can view the non-strict variant of the dynamics as falling between the strict dynamics and node-based threshold models, where the existence and structural patterns of two-node hyperedges are key to determining whether the non-strict dynamics behave more like strict or node-based threshold dynamics⁴.

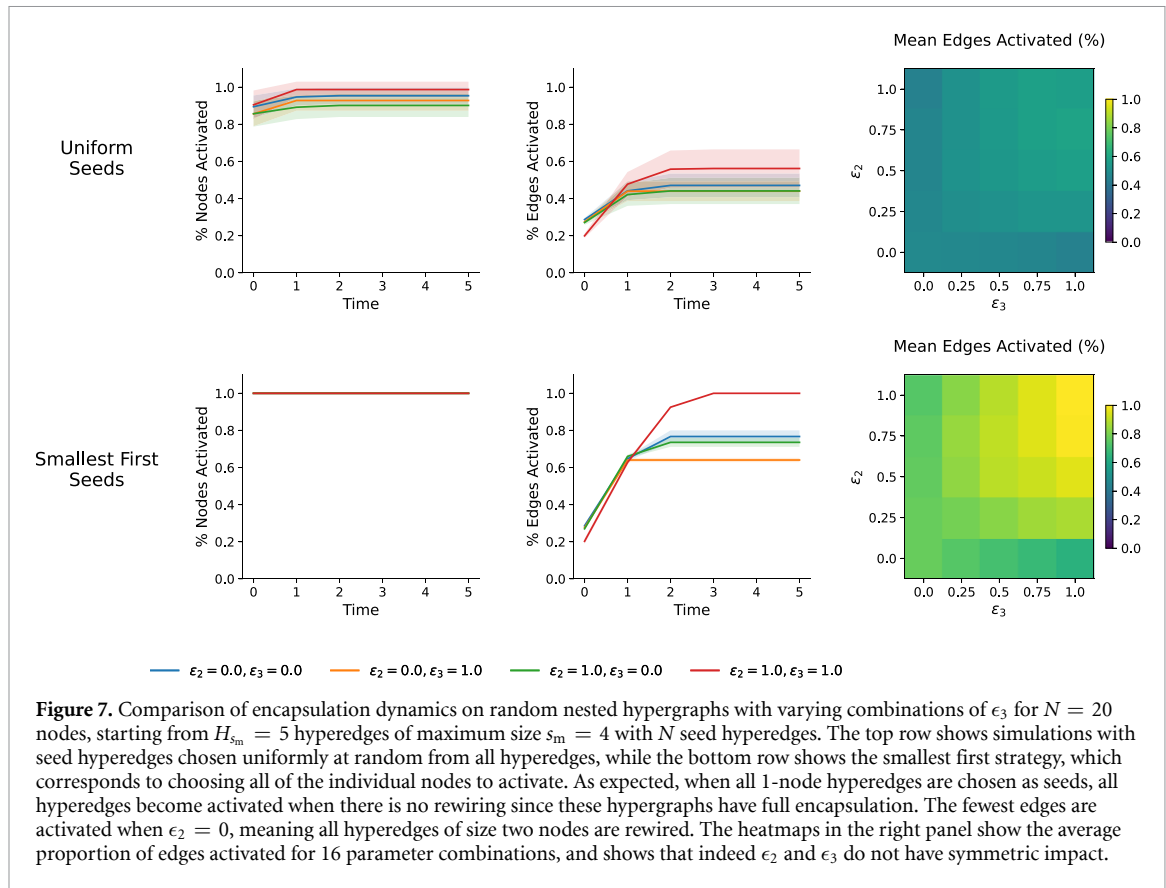
We simulate encapsulation dynamics by constructing the encapsulation DAG, but only keeping edges between hyperedges at adjacent layers, i.e. where the difference in size is one. In our simulations, we first place a given number of seed-activated hyperedges using one of the strategies described below. We then count for each hyperedge how many of its encapsulated hyperedges are seeds and deterministically simulate the dynamics forward. After each iteration, for every hyperedge α with size s_α nodes we update the number of its encapsulated $s_\alpha - 1$ hyperedges that became activated. In practice, it is more efficient to update these counts by maintaining a reverse adjacency list of the encapsulation DAG so that we need only loop over the newly activated hyperedges and update the counts for the inactive hyperedges that they are encapsulated by.

We consider four different strategies for choosing seed hyperedges:

- **Uniform:** choose hyperedges uniformly at random.
- **Size Biased:** choose hyperedges with probability proportional to their size (i.e. choose the largest hyperedges first).
- **Inverse Size Biased:** choose hyperedges with probability proportional to their inverse size (i.e. choose the smallest hyperedges first).

³ Inspired by the language of topology, we may also call these dynamics *subface* dynamics, referring to the fact that a subface of a simplicial complex would need to be activated for a larger face to activate.

⁴ We also report simulations using more traditional threshold contagion dynamics based on node activations in appendix D.



- **Smallest First:** explicitly choose the smallest hyperedges first. Practically, arrange the hyperedges in a vector ordered by increasing size, with hyperedges of the same size in random order. Choose seed hyperedges starting from the beginning of this vector.

We expect that in a hypergraph with deep encapsulation relationships the smallest first seeding strategy will be the most effective for strict encapsulation dynamics, since the small hyperedges must be activated or the dynamics will never reach the entire structure. In contrast, in non-strict encapsulation dynamics it may be the case that activating the largest hyperedges first will activate the most nodes that will in turn activate many pairwise hyperedges, potentially leading to more activation overall.

4.1. Simulations on the RNHM

In figure 7 we compare the encapsulation dynamics on random nested hypergraphs with varying combinations of ϵ_3 for RNHM parameters $N = 20$, $s_m = 4$, $H_{s_m} = 5$. In these simulations, we also include all of the individual nodes in the hypergraph. We show results using both uniform (top row) and smallest first (bottom row) seeding strategies, with number of seeds the same as the number of nodes N . Each point is an average over 50 realizations of the hypergraph and 50 simulations per realization. The smallest first strategy is more effective for all parameters, consistent with the ‘campfire’ intuition of lighting the fuel to burn the logs.

In the smallest first simulations, all hyperedges are activated consistently when there is no rewiring of any hyperedges ($\epsilon_2 = \epsilon_3 = 1$, red line in figure 7), as expected. Interestingly, the dynamics are qualitatively different when either the 2- or 3-node hyperedges are rewired, but the other is left alone. More hyperedges are activated when only three-node hyperedges are rewired ($\epsilon_2 = 1, \epsilon_3 = 0$, green line) compared to when only two-node hyperedges are rewired ($\epsilon_2 = 0, \epsilon_3 = 1$, orange line). However, it is not the case that the most rewiring leads to the slowest activation dynamics. We attribute this to a combination of the stochasticity of the rewiring process, our imposition of connectivity on the hypergraphs, and the relatively small number of nodes N , which can lead to situations where rewired hyperedges encapsulate each other randomly (see the encapsulation DAG in black for $\epsilon_2 = 0, \epsilon_3 = 0$ in figure 6, for example).

We also note that the smallest first seeding strategy as used in this setting would make node-based threshold dynamics trivial, since every node is activated in the seeding process. This illustrates the key conceptual difference between node-based and encapsulation-based dynamics: the latter requires explicit higher-order coordination among activated nodes in the form of encapsulation in the hypergraph structure.

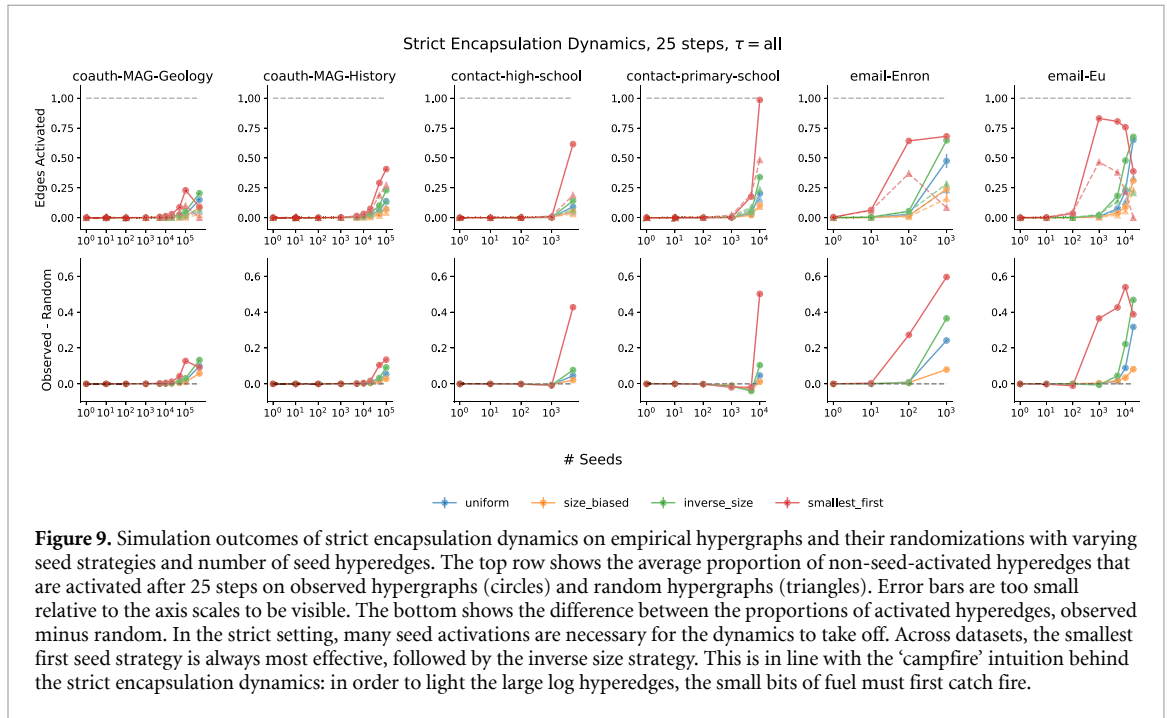
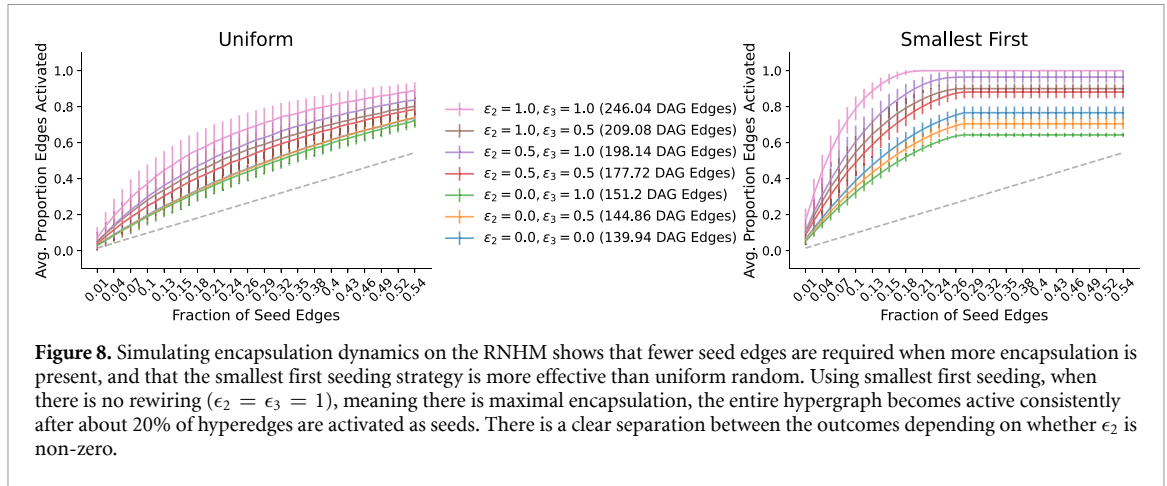


Figure 8 shows the average outcome of simulations on RNHMs with an increasing number of seed hyperedges again chosen with either uniform or smallest first strategy (25 realizations, 100 simulations per realization). In both cases there appear to be two distinct trends in the encapsulation dynamics results corresponding to the total number of DAG edges (reported in the legend) and depending on whether ϵ_2 is zero, meaning all two-node hyperedges are rewired. Activation spreads to a larger number of hyperedges when $\epsilon_2 > 0$, consistent with the result from figure 7. When two-node hyperedges are fully rewired, even with 50% of edges being activated as seeds, only about 75% of the total edges are activated by the end of the process in the best case.

4.2. Simulations on empirical data

We also simulated the encapsulation dynamics on the same empirical datasets described in table 1 and their randomizations. In the top rows of figures 9 and 10, we show the proportion of non-seed hyperedges activated after 25 steps across all datasets with varying seed strategies and increasing number of initially active seed hyperedges⁵. In the bottom rows of each figure, we show the difference between the observed and randomized outcomes.

⁵ Since the dynamics are deterministic once the seed hyperedges are chosen, usually only a small number of simulation steps are needed before the spreading stops. 25 steps is more than necessary for all of these datasets.

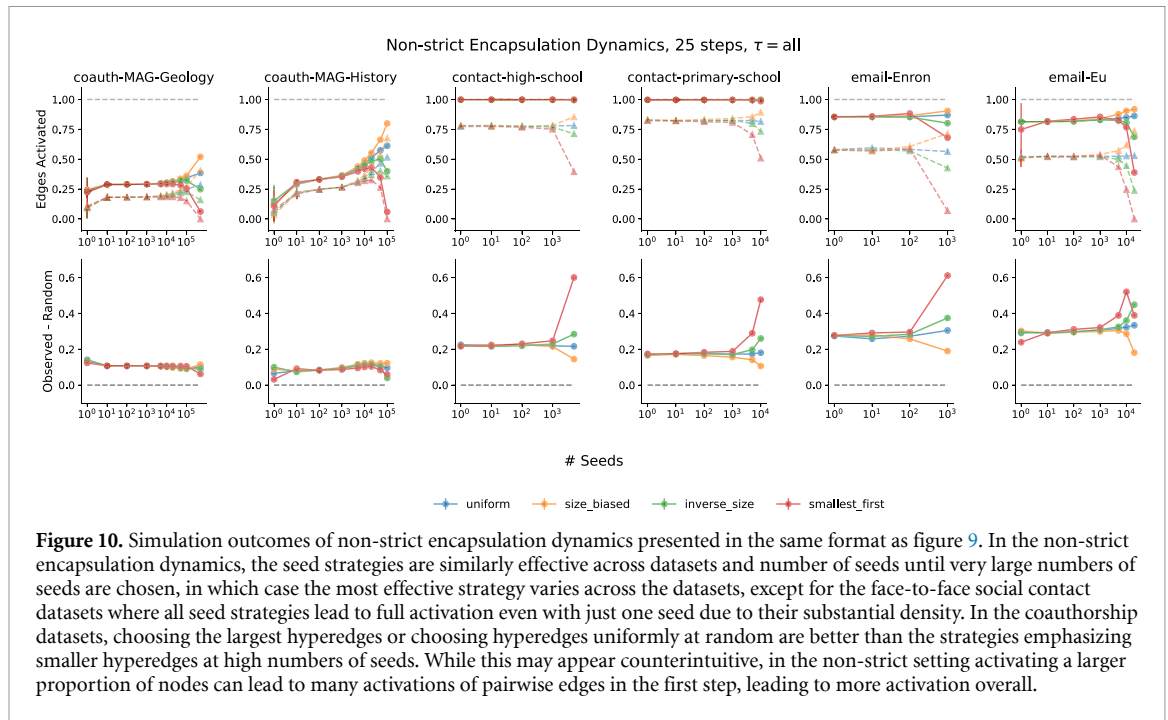


Figure 10. Simulation outcomes of non-strict encapsulation dynamics presented in the same format as figure 9. In the non-strict encapsulation dynamics, the seed strategies are similarly effective across datasets and number of seeds until very large numbers of seeds are chosen, in which case the most effective strategy varies across the datasets, except for the face-to-face social contact datasets where all seed strategies lead to full activation even with just one seed due to their substantial density. In the coauthorship datasets, choosing the largest hyperedges or choosing hyperedges uniformly at random are better than the strategies emphasizing smaller hyperedges at high numbers of seeds. While this may appear counterintuitive, in the non-strict setting activating a larger proportion of nodes can lead to many activations of pairwise edges in the first step, leading to more activation overall.

In strict encapsulation dynamics (figure 9), where pairwise edges can only be activated if one of their constituent nodes is present as a hyperedge, no further hyperedges are activated on average for small numbers of seeds across the coauthorship and face-to-face contact datasets. In the email datasets, the dynamics already take off with just ten seed hyperedges and the smallest hyperedges first strategy clearly has an advantage in both the observed and randomized datasets. In fact, across all of the datasets the smallest first strategy is the most effective, and it also tends to be the strategy with largest difference in final activations between the observed and layer randomized hypergraphs. In general, activations on the layer randomization are much lower than in the observed hypergraphs, which is as expected since the observed data contains many more encapsulation relationships.

In the non-strict encapsulation dynamics (figure 10), we again see that more non-seed edges are activated in the observed hypergraph with more encapsulation relationships. In the face-to-face social contact datasets, a single seed is enough to activate the entire observed hypergraph. Similarly, in the email datasets the final number of activations is consistent regardless of the number of seeds, until falling off at high numbers of seeds, likely due to the smaller proportion of available hyperedges to activate. However, in the layer-randomized hypergraphs, in the face-to-face contact and email datasets there appears to be a limit on the amount of non-seed hyperedges that can become activated.

We also note that in non-strict encapsulation dynamics, there is not a clearly best hyperedge seed placement strategy across the datasets. It is intuitive that the size biased strategies work well in non-strict dynamics with small numbers of seeds, since this strategy will by definition activate the most nodes, and these nodes can in turn activate pairwise edges they participate in, essentially translating into more seeds.

5. Conclusion

Higher-order networks have emerged in recent years as a promising approach to represent and model interacting systems. Among this broad family of models, approaches based on hypergraphs help to characterise the global structure and collective dynamics when interactions involve more than two agents. In this work, we have proposed novel ways to quantify the relations between hyperedges in real-world datasets. Based on the notions of overlap and encapsulation, we propose two alternative ways to represent a hypergraph as a graph where the nodes are the original hyperedges. In this *line graph* representation, edges may be directed to encode the encapsulation of a hyperedge in another, or undirected to encode the number of nodes in common between them. We have focused in detail on the structure induced by encapsulation, proposing a randomization strategy to erase encapsulation relations between hyperedges, while preserving other structural patterns, and quantifying how different real-world data are from what would be expected in a simplicial complex representation.

As a second step, we turned to dynamics. In contrast with works focusing on the difference exhibited by a dynamical process on a hypergraph and on its corresponding projection on a graph, we explore the impact of encapsulation on spreading and compare the dynamics taking place on real-work hypergraphs and their randomization. To do so, we focus on a dynamical process specifically designed for hypergraphs—the encapsulation dynamics is trivial on graphs—and demonstrate that encapsulation facilitates spreading in situations when smaller hyperedges fuel the activation of larger hyperedges. Our work contributes to the recent efforts to understand how hypergraph structure impacts dynamics. Future research directions include a more thorough focus on the importance of overlap, but also testing our metrics to study other dynamical models, e.g. for synchronisation.

There remain many potential avenues for future work in this area. We have focused on a simple, size-layer-based approach to randomizing hypergraphs, but there exist in the literature other ways of randomizing hyperedges, including the configuration model approach introduced in [11] and the multiplex approaches in [45, 60]. In contrast to our randomization, which preserves the size distribution of hyperedges and the unlabeled within-layer node degree distributions, these models preserve more general notions of degree, including the overall hyperdegree and the detailed within-layer degree of each node. Another potential research direction concerns the encapsulation dynamics, that was kept as simple as possible for the purpose of this work, but could be defined in different variants, as we allude to in appendix C, in the same way that different types of threshold dynamical models have been explored in the literature. Finally the intersection and encapsulation relations are just two out of the several ways in which the relation between hyperedges can be measured. A combined analysis of the multiple line graphs that can be associated to the same hypergraph is also a promising research direction.

In this work, we ignored the temporal aspect of hypergraphs, however in the future the ideas introduced here could be extended to understand encapsulation patterns in temporal or dynamic hypergraphs, following work such as [25]. Our work could also be integrated with existing literature on higher-order motifs in hypergraphs [39, 40]. Further research could also be done on analyzing the DAG structures we investigated here using recent work on the cyclic analysis of DAGs [61].

Data availability statement

Code implementing the measurements and simulations shown in this paper will be made available at <https://github.com/tlarock/encapsulation-dynamics/> [36]. All of the empirical data was made available with the publication of [5] and can be found online at www.cs.cornell.edu/~arb/data/.

Acknowledgments

The authors acknowledge support from the EPSRC Grant EP/V03474X/1. We also thank the anonymous reviewer for their helpful comments. T L acknowledges the use of open source code made available by the developers of many projects including NumPy [22], SciPy [62], NetworkX [21], Matplotlib [24], and complex Group Interactions (XGI) [33].

Appendix A. Construction of overlap and encapsulation structures

In this appendix we describe the algorithms we use for constructing overlap and encapsulation structures. We use the same basic procedure for each and give pseudocode for this procedure in algorithm 1. We first assign each hyperedge e a unique label $\text{map}_e \in 1, \dots, |E|$ and construct a node-membership lookup table memb_u between each node and the list of hyperedges it participates in (lines 2–7). We then loop over each hyperedge $\alpha \in E$, and for each node $u \in \alpha$ we add edges from α to other hyperedges $\beta \in \text{memb}_u$ in which u participates based on the relation we are interested in (lines 8–12). The relation determines the logic implemented in line 12. For the intersection graph, we add edges to L from hyperedge α to other hyperedges $\beta \in \text{memb}_u$ if the intersection $\alpha \cap \beta$ is non-empty, potentially including the weight defined in the main text. For the encapsulation DAG, we only add edges to hyperedges β that are encapsulated by α , meaning we add edges where $\beta \subset \alpha$ (or *vice-versa*). After repeating this loop for each node in α , the out-neighbors of α in L represent all of the hyperedges $\beta \in E$ that have the relevant relationship with α .

Algorithm 1. Given a hypergraph $H = (V, E)$, construct a line graph.

```

1: procedure LINEGRAPH( $H$ )
2:    $\text{label} \leftarrow 1, \text{map}_e \leftarrow \emptyset, \text{memb}_u \leftarrow \emptyset$ 
3:   for  $\alpha \in E$  do
4:      $\text{map}_\alpha \leftarrow \text{label}$ 
5:      $\text{label} += 1$ 
6:     for  $u \in \alpha$  do
7:        $\text{memb}_u \leftarrow \text{memb}_u \cup \text{map}_\alpha$ 
8:    $L \leftarrow \emptyset$ 
9:   for  $\alpha \in E$  do
10:    for  $u \in \alpha$  do
11:      for  $\beta \in \text{memb}_u$  do
12:        Decide whether to add edge between  $\alpha, \beta$  to  $L$ 
  return  $L$ 

```

The complexity of this construction has two terms corresponding to the two nested for loops. The first doubly nested loop runs over all $m = |E|$ hyperedges to construct a mapping from hyperedges to labels, then all of the nodes in each hyperedge to fill the mapping from nodes to the hyperedges in which they appear. This operation takes $O(m \cdot s_{\max})$ time, where $s_{\max} = \max_{e \in E} s_e$ is the maximum size of a hyperedge. Once the mappings are constructed, we enter the triply nested loop to find encapsulation and overlap relationships. The worst case time for an inner loop is the size of the largest hyperedge s_{\max} times the highest degree node $k_{\max} = \max_{u \in V} |\{e | u \in e; e \in E\}|$. Clearly the triply nested loop dominates the doubly nested loop and so the worst case running time is $O(m \cdot s_{\max} \cdot k_{\max})$.

A.1. Constructing encapsulation DAGs with the set-trie

The triply nested for loop in algorithm 1 is a brute force method over the data structures we have constructed. Therefore, the above running time is an upper-bound on the most efficient algorithm. One potentially more efficient alternative is to construct an encapsulation DAG by augmenting the construction of a set-trie data structure [54]. Here we give an informal description of one implementation of this strategy and its complexity.

Assume we are given a list of hyperedges E on nodes $V = 1, \dots, N$ in partial order by increasing size (i.e. the smallest hyperedges first) where each hyperedge e is represented as a tuple $e = (u_1 < u_2 < \dots < u_{s_e})$ sorted in increasing order by node label. The algorithm constructs a set-trie by looping over the hyperedges e in increasing order of size, first running the `GetAllSubsets` procedure specified in [54] on e , then inserting e into the set-trie. Since we are inserting the hyperedges in order of their size, we can be sure that all possible subsets of e already exist in the data structure before e is inserted, thus `GetAllSubsets` returns all DAG edges for each hyperedge e .

We leave detailed analysis of this alternative algorithm for future research, but note here that the expected size w_e of the search tree for each input hyperedge e is

$$\mathbb{E}(w_e) \leq \prod_{u \in e} \left[\frac{2 + q^{2^{N-i-1}}}{1 + q^{2^{N-i-1}}} \right],$$

where q is an assumed fixed probability of a hyperedge to be excluded from a set of hyperedges over V , i.e. the probability of inclusion in the hypergraph is assumed to be $P[e \in E] = p$ and $q = 1 - p$ (see [54] sections 4.2 and 4.3.1 for details). Thus an informal bound can be stated as $O(m(s_{\max} + \mathbb{E}(w_e)))$. That is, for each of the m hyperedges it takes worst case s_{\max} operations to insert into the set-trie and a depth-first search over a tree with expected size $\mathbb{E}(w_e)$ to find subsets of e .

Future work could formalize and refine this running time as well as give strategies for optimizing the node ordering as a preprocessing step and pruning the search space to avoid redundant computations, among other optimizations. For example, it would be trivial to take advantage of the incomplete DAG when searching for subsets of larger hyperedges, since it is in effect a memoization of previous subset searches. We also expect that an optimized node ordering will reduce the effect of k_{\max} on the running time, leading to an advantage over the original algorithm. However, these optimizations are likely to be complex to analyze in a general way, so we have considered them outside of the scope of this paper. Finally, we note that here we have only described how to use the set-trie to find encapsulation relationships; future work could attempt to extend this to efficiently find overlap relationships as well.

Table B1. Number of nodes, hyperedges, and DAG edges in each dataset after removing multiedges. In our measurements we include all hyperedges, while in simulations we focus on the largest connected components.

Dataset	n	m	Proj. density	DAG edges
coauth-MAG-Geology	1256 385	1203 895	10^{-5}	1666 414
coauth-MAG-History	1014 734	895 439	10^{-5}	276 588
contact-high-school	327	7818	0.11	7942
contact-primary-school	242	12 704	0.29	16 199
email-Enron	143	1512	0.18	8240
email-Eu	998	25 027	0.06	277 224

Appendix B. Data

Table B1 shows the same statistics as table 1, but for the whole hypergraph, rather than just the largest connected component.

Appendix C. Discussion of alternative dynamics

Due to the multidimensionality inherent to hypergraphs, there are numerous valid choices for specifying a spreading process of the type we study here, each of which have their own conceptual and practical advantages and pitfalls. In this appendix we discuss some of the potential alternatives that could be investigated in the future. We focus specifically on the specification of spreading over hyperedges—for a brief discussion of node-based threshold models on hypergraphs, see appendix D.

The first and most important choice in specifying the dynamics is deciding which hyperedges can influence one another. In the main text, we presented a model where only hyperedges at adjacent levels in the encapsulation DAG can influence each other, e.g. one in which only hyperedges of size $k - 1$ can influence a hyperedge of size k . These are in some sense the most directly applicable to the ‘ideal’ encapsulation DAG, since the dynamics directly spread over the DAG structure. However, we are also interested in how our spreading process unfolds on empirical hypergraphs, and we cannot know in advance whether the DAG connectivity will be suitable for spreading.

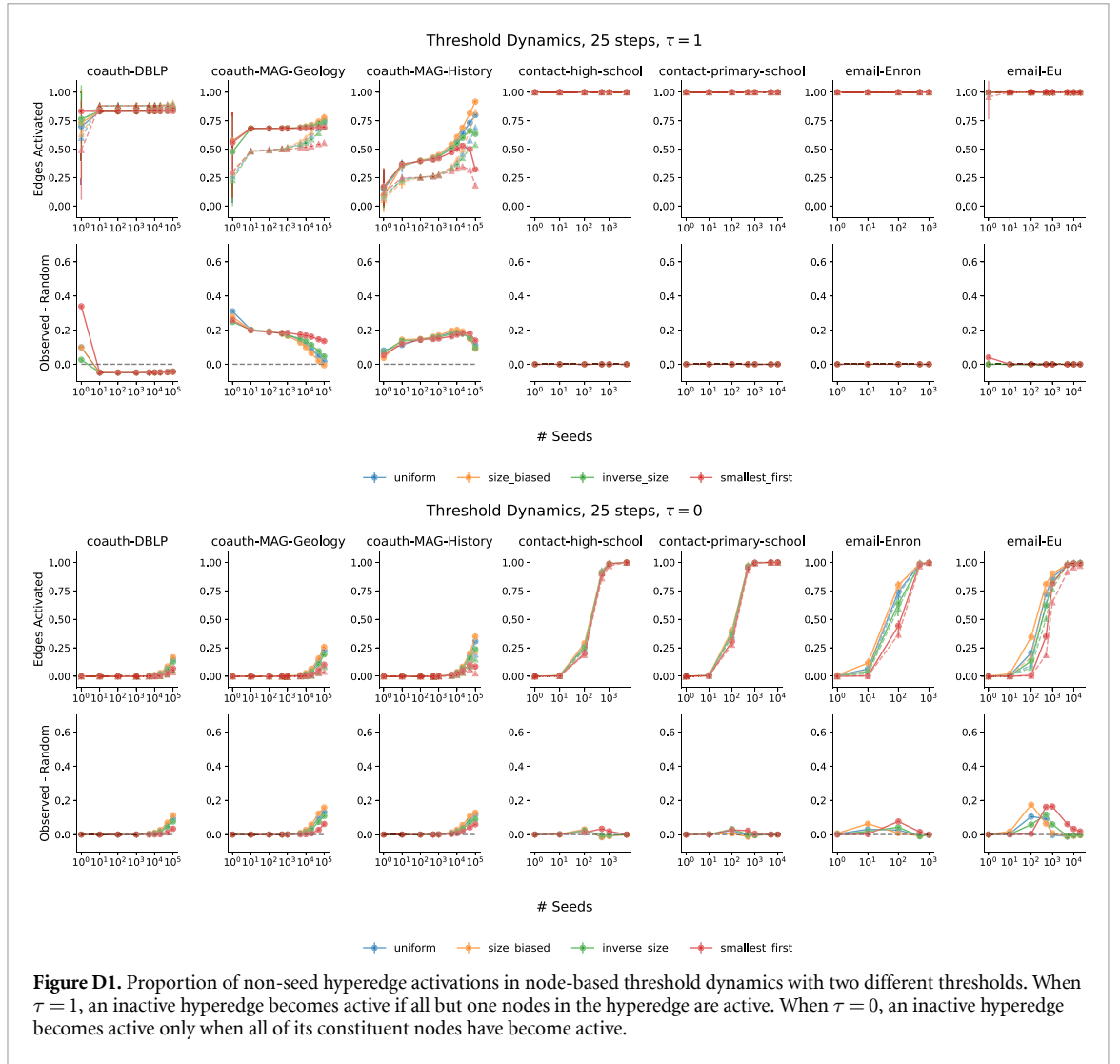
With this limitation in mind, we can also specify a version of encapsulation dynamics where we relax the condition from requiring immediately adjacent hyperedges to *empirically adjacent* hyperedges, meaning that a hyperedge α can be influenced by hyperedges it encapsulates that are of the maximum size $k < |\alpha|$ existing in the hypergraph. For example, if a hyperedge on four nodes does not encapsulate any hyperedges on three nodes, but does encapsulate a hyperedge on two nodes, we allow this smaller hyperedge to influence the larger.

The encapsulation dynamics presented in the main text are the most true to the spirit of the encapsulation relation, since they require that the encapsulation DAG has a specific structure. The empirical encapsulation relaxation is more flexible and compatible with the variety of structures we expect to see in empirical data, but the cost of this flexibility is that in some cases very small hyperedges—including individual nodes in the coauthorship setting—can ‘punch above their weight’ by activating much larger hyperedges just by virtue of being the only observed encapsulated edge.

We can address this issue in a few ways. In the first place, we could set the threshold τ to be at least the number of individual nodes in the hyperedge. With this threshold, it would only be possible for single nodes to activate a larger hyperedge if all of them were activated. However, this ‘global’ threshold could have the effect of making it impossible to activate some hyperedges, for example a hyperedge with only one encapsulation of size $k - 1$, which would also be counter-intuitive. Instead, size-specific threshold models could be given, such that a different number of different sized hyperedges could be necessary to activate a hyperedge. Alternatively, more complex (potentially stochastic) threshold functions could be specified that control the influence of hyperedges of different sizes in determining the activation threshold of a hyperedge (see for example [42]).

There is also the question of whether activation should go in only one direction, from smaller hyperedges up to larger hyperedges, or in both directions. In this work we have only allowed activation to flow from smaller to larger hyperedges, with the single exception of the non-strict dynamics, where node activation can influence pairwise interactions. However, it would be equally reasonable to assume that once a larger hyperedge has been activated, all or some of its subsets also become active.

Finally, more complex processes that allow hyperedges to deactivate and reactivate could be studied. Our threshold model is in the flavor of a Susceptible-Infectious contagion, and so a logical next step is to extend the dynamics to study more complex dynamics like Susceptible-Infectious-Susceptible,



Susceptible-Infectious-Recovered, etc. We leave investigation of these different dynamical model specifications for future work.

Appendix D. Threshold contagion model

In this appendix, we show some results on a traditional node-based threshold contagion model on a hypergraph to contrast with the encapsulation dynamics we introduced in the main text. Just as in encapsulation dynamics, in our threshold model every node $u \in V$ and hyperedge $\alpha \in E$ in the hypergraph is in a binary state, either inactive or active, in each discrete timestep. At each step, an inactive hyperedge α , $x_\alpha = 0$ is activated if the number of already-activated nodes within the hyperedge is larger than a threshold. When a hyperedge is activated, all of its member nodes $u \in \alpha$ are also activated. We define the threshold based on the size of the hyperedge, specifically $|\alpha| - \tau$. An inactive hyperedge α will be activated if

$$\sum_{u \in \alpha} s_u \geq |\alpha| - \tau,$$

that is, if the number of activated nodes is greater than the size of the hyperedge minus the threshold.

These dynamics could still be sensitive to encapsulation structure in a hypergraph, however the overlap structure of the hypergraph can play an equally important role, since there is no requirement that smaller hyperedges are activated first to activate enough nodes to finally activate larger hyperedges.

We run simulations on empirical datasets using two threshold values: $\tau = 0$ and $\tau = 1$ and present the results in figure D1. When $\tau = 1$ (top plot), meaning that an inactive hyperedge α becomes active when the number of inactive nodes remaining in α is 1, a single seed activates the entire hypergraph for both the face-to-face contact and email datasets. In the coauthorship datasets, full activation is never achieved in either observed or randomized datasets.

When $\tau = 0$, meaning all nodes must be activated for a hyperedge to become active, a sort of unanimity condition, the outcomes are dependent on the dataset. Starting with the email-Eu dataset, we see that as the number of seed hyperedges increases, choosing the largest hyperedges first is the most effective strategy on the observed data until the number of seeds increases passed 10^3 , where all of the methods converge. In the email-Enron dataset there is a similar pattern, but the difference between the outcome on the observed hypergraph and the random hypergraph is smaller across the simulations. The two contact datasets show similar patterns across all of the seeding strategies and in both observed and randomized hypergraphs, with full activation being achieved for the largest numbers of seeds. Finally, in the coauthorship datasets almost no activation occurs until more than 10^4 hyperedges are activated as seeds, and choosing seeds proportional to their size is the best strategy.

ORCID iDs

Timothy LaRock  <https://orcid.org/0000-0003-0801-3917>

Renaud Lambiotte  <https://orcid.org/0000-0002-0583-4595>

References

- [1] Aksoy S G, Joslyn C, Marrero C O, Praggastis B and Purvine E 2020 Hypernetwork science via high-order hypergraph walks *EPJ Data Sci.* **9** 16
- [2] Baccini F, Geraci F and Bianconi G 2022 Weighted simplicial complexes and their representation power of higher-order network data and topology *Phys. Rev. E* **106** 034319
- [3] Battiston F *et al* 2021 The physics of higher-order interactions in complex systems *Nat. Phys.* **17** 1093–8
- [4] Benson A R 2019 Three hypergraph eigenvector centralities *SIAM J. Math. Data Sci.* **1** 293–312
- [5] Benson A R, Abebe R, Schaub M T, Jadbabaie A and Kleinberg J 2018 Simplicial closure and higher-order link prediction *Proc. Natl Acad. Sci.* **115** E11221–30
- [6] Bick C, Böhle T and Kuehn C 2022 Multi-population phase oscillator networks with higher-order interactions *Nonlinear Differ. Equ. Appl.* **29** 64
- [7] Bick C, Gross E, Harrington H A and Schaub M T 2023 What are higher-order networks? *SIAM Rev.* **65** 686–731
- [8] Bodó A, Katona G Y and Simon P L 2016 SIS epidemic propagation on hypergraphs *Bull. Math. Biol.* **78** 713–35
- [9] Cencetti G, Battiston F, Lepri B and Karsai M 2021 Temporal properties of higher-order interactions in social networks *Sci. Rep.* **11** 7028
- [10] Cencetti G, Contreras D A, Mancastroppa M and Barrat A 2023 Distinguishing simple and complex contagion processes on networks *Phys. Rev. Lett.* **130** 247401
- [11] Chodrow P S 2020 Configuration models of random hypergraphs *J. Complex Netw.* **8** cnaa018
- [12] Civilini A, Sadekar O, Battiston F, Gómez-Gardeñes J and Latora V 2023 Explosive cooperation in social dilemmas on higher-order networks (arXiv:2303.11475)
- [13] Costa L d F 2021 Further generalizations of the jaccard index (arXiv:2110.09619)
- [14] Courtney O T and Bianconi G 2016 Generalized network structures: the configuration model and the canonical ensemble of simplicial complexes *Phys. Rev. E* **93** 062311
- [15] Courtney O T and Bianconi G 2017 Weighted growing simplicial complexes *Phys. Rev. E* **95** 062301
- [16] de Arruda G F, Petri G and Moreno Y 2020 Social contagion models on hypergraphs *Phys. Rev. Res.* **2** 023032
- [17] Do M T, Yoon S-e, Hooi B and Shin K 2020 Structural patterns and generative models of real-world hypergraphs *Proc. 26th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (ACM)* pp 176–86
- [18] Evans T S 2010 Clique graphs and overlapping communities *J. Stat. Mech.* **P** 12037
- [19] Evans T S and Lambiotte R 2009 Line graphs, link partitions and overlapping communities *Phys. Rev. E* **80** 016105
- [20] Guillebeault D, Becker J and Centola D 2018 Complex contagions: a decade in review *Complex Spreading Phenomena in Social Systems. Computational Social Sciences* ed S Lehmann and Y Y Ahn (Springer) (https://doi.org/10.1007/978-3-319-77332-2_1)
- [21] Hagberg A A, Schult D A and Swart P J 2008 Exploring network structure, dynamics and function using networkx *Proc. 7th Python in Science Conf.*, ed G Varoquaux, T Vaught and J Millman pp 11–15
- [22] Harris C R *et al* 2020 Array programming with NumPy *Nature* **585** 357–62
- [23] Higham D J and De Kergorlay H-L 2021 Epidemics on hypergraphs: spectral thresholds for extinction *Proc. R. Soc. A* **477** 20210232
- [24] Hunter J D 2007 Matplotlib: a 2D graphics environment *Comput. Sci. Eng.* **9** 90–95
- [25] Iacopini I, Karsai M and Barrat A 2023 The temporal dynamics of group interactions in higher-order social networks (arXiv:2306.09967)
- [26] Iacopini I, Petri G, Barrat A and Latora V 2019 Simplicial models of social contagion *Nat. Commun.* **10** 2485
- [27] Kim J, Lee D-S and Goh K-I 2023 Contagion dynamics on hypergraphs with nested hyperedges (<https://doi.org/10.1103/PhysRevE.108.034313>)
- [28] Klamt S, Haus U-U and Theis F 2009 Hypergraphs and cellular networks *PLoS Comput. Biol.* **5** e1000385
- [29] Lambiotte R, Rosvall M and Scholtes I 2019 From networks to optimal higher-order models of complex systems *Nat. Phys.* **15** 313–20
- [30] Lambiotte R and Schaub M T 2021 *Modularity and Dynamics on Complex Networks* (Cambridge University Press)
- [31] Lanchier N and Neuffer J 2013 Stochastic dynamics on hypergraphs and the spatial majority rule model *J. Stat. Phys.* **151** 21–45
- [32] Landry N W, Amburg I, Shi M and Aksoy S G 2023 Filtering higher-order datasets (arXiv:2305.06910)
- [33] Landry N W, Lucas M, Iacopini I, Petri G, Schwarze A, Patania A and Torres L 2023 XGI: a Python package for higher-order interaction networks *J. Open Source Softw.* **8** 5162
- [34] Landry N W and Restrepo J G 2023 Opinion disparity in hypergraphs with community structure *Phys. Rev. E* **108** 034311
- [35] Landry N W, Young J-G and Eikmeier N 2023 The simpliciality of higher-order networks (arXiv:2308.13918)
- [36] LaRock T 2023 Encapsulation dynamics github repository (available at: <https://github.com/tlarock/encapsulation-dynamics>)

- [37] Lee G, Choe M and Shin K 2021 How do hyperedges overlap in real-world hypergraphs?—patterns, measures and generators *Proc. Web Conf. 2021* (ACM) pp 3396–407
- [38] Leskovec J, Kleinberg J and Faloutsos C 2007 Graph evolution: densification and shrinking diameters *ACM Trans. Knowl. Discovery Data* **1** 2
- [39] Lotito Q F, Musciotto F, Battiston F and Montresor A 2023 Exact and sampling methods for mining higher-order motifs in large hypergraphs (<https://doi.org/10.1007/s00607-023-01230-5>)
- [40] Lotito Q F, Musciotto F, Montresor A and Battiston F 2022 Higher-order motif analysis in hypergraphs *Commun. Phys.* **5** 79
- [41] Lotito Q F, Musciotto F, Montresor A and Battiston F 2023 Hyperlink communities in higher-order networks (arXiv:2303.01385)
- [42] Lucas M, Iacopini I, Robiglio T, Barrat A and Petri G 2023 Simplicially driven simple contagion *Phys. Rev. Res.* **5** 013201
- [43] Malizia F, Lamata-Otín S, Frasca M, Latora V and Gómez-Gardeñes J 2023 Hyperedge overlap drives explosive collective behaviors in systems with higher-order interactions (arXiv:2307.03519)
- [44] Mastrandrea R, Fournet J and 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** e0136497
- [45] Musciotto F, Battiston F and Mantegna R N 2021 Detecting informative higher-order interactions in statistically validated hypergraphs *Commun. Phys.* **4** 218
- [46] Neuhäuser L, Mellor A and Lambiotte R 2020 Multibody interactions and nonlinear consensus dynamics on networked systems *Phys. Rev. E* **101** 032310
- [47] Neuhäuser L, Scholkemper M, Tudisco F and Schaub M T 2023 Learning the effective order of a hypergraph dynamical system (arXiv:2306.01813)
- [48] Newman M 2018 *Networks* (Oxford University Press)
- [49] Noonan J and Lambiotte R 2021 Dynamics of majority rule on hypergraphs *Phys. Rev. E* **104** 024316
- [50] Patania A, Petri G and Vaccarino F 2017 The shape of collaborations *EPJ Data Sci.* **6** 1–16
- [51] Petri G and Barrat A 2018 Simplicial activity driven model *Phys. Rev. Lett.* **121** 228301
- [52] Petri G, Expert P, Turkheimer F, Carhart-Harris R, Nutt D, Hellyer P J and Vaccarino F 2014 Homological scaffolds of brain functional networks *J. R. Soc. Interface* **11** 20140873
- [53] Salnikov V, Cassese D and Lambiotte R 2018 Simplicial complexes and complex systems *Eur. J. Phys.* **40** 014001
- [54] Savnik I, Akulich M, Krnc M and Škrekovski R 2021 Data structure set-trie for storing and querying sets: theoretical and empirical analysis *PLoS One* **16** e0245122
- [55] Schaub M T, Benson A R, Horn P, Lippner G and Jadbabaie A 2020 Random walks on simplicial complexes and the normalized hodge 1-laplacian *SIAM Rev.* **62** 353–91
- [56] Sharma A 2020 Hypergraph analytics: modeling higher-order structures and probabilities *PhD Thesis* University of Minnesota
- [57] Sharma A, Moore T J, Swami A and Srivastava J 2017 Weighted simplicial complex: a novel approach for predicting small group evolution *Advances in Knowledge Discovery and Data Mining* vol 10234, ed J Kim, K Shim, L Cao, J-G Lee, X Lin and Y-S Moon (Springer International Publishing) pp 511–23
- [58] Sinha A, Shen Z, Song Y, Ma H, Eide D, Hsu B-J P and Wang K 2015 An overview of microsoft academic service (MAS) and applications *Proc. 24th Int. Conf. on World Wide Web* (ACM Press)
- [59] Stehlé J *et al* 2011 High-resolution measurements of face-to-face contact patterns in a primary school *PLoS One* **6** e23176
- [60] Sun H and Bianconi G 2021 Higher-order percolation processes on multiplex hypergraphs *Phys. Rev. E* **104** 034306
- [61] Vasiliauskaite V, Evans T S and Expert P 2022 Cycle analysis of directed acyclic graphs *Physica A* **596** 127097
- [62] Virtanen P *et al* (SciPy 1.0 contributors) 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python *Nat. Methods* **17** 261–72
- [63] Watts D J 2002 A simple model of global cascades on random networks *Proc. Natl Acad. Sci.* **99** 5766–71
- [64] Watts D J and Strogatz S H 1998 Collective dynamics of ‘small-world’ networks *Nature* **393** 440–2
- [65] Yin H, Benson A R, Leskovec J and Gleich D F 2017 Local higher-order graph clustering *Proc. 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (ACM Press) pp 555–64
- [66] Zhang Y, Lucas M and Battiston F 2023 Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes *Nat. Commun.* **14** 1605