The Artificial Benchmark for Community Detection with Outliers and Overlapping Communities (ABCD+o^2)

Jordan Barrett[*], Ryan DeWolfe[†], Bogumił Kamiński[‡], Paweł Prałat[§], Aaron Smith[¶], François Théberge[‖]

March 16, 2026

Abstract

The **A**rtificial **B**enchmark for **C**ommunity **D**etection (**ABCD**) graph is a random graph model with community structure and power-law distribution for both degrees and community sizes. The model generates graphs similar to the well-known **LFR** model but it is faster, more interpretable, and can be investigated analytically. In this paper, we use the underlying ingredients of the **ABCD** model, and its generalization to include outliers (**ABCD+o**), and introduce **ABCD+o^2**, another variant that allows for overlapping communities.

Keywords— ABCD, community detection, overlapping communities, benchmark models

1 Introduction

One of the most important features of real-world networks is their community structure, as it reveals the internal organization of nodes. In social networks, communities may represent groups by interest; in citation networks, they correspond to related papers; in the Web graph, communities are

[*]Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada; e-mail: jbarrett@dal.ca

[†]Department of Mathematics, Toronto Metropolitan University, Toronto, ON, Canada; e-mail: ryan.dewolfe@torontomu.ca

[‡]Decision Analysis and Support Unit, SGH Warsaw School of Economics, Warsaw, Poland; e-mail: bkamins@sgh.waw.pl

[§]Department of Mathematics, Toronto Metropolitan University, Toronto, ON, Canada; e-mail: pralat@torontomu.ca

[¶]Department of Mathematics and Statistics, University of Ottawa, Ottawa, ON, Canada; e-mail: asmi28@uOttawa.ca

[‖]Tutte Institute for Mathematics and Computing, Ottawa, ON, Canada; email: theberge@ieee.org

formed by pages on related topics, etc. Identifying communities in a network is therefore valuable, as it helps us understand the structure of the network.

Detecting communities is quite a challenging task. In fact, there is no definition of community that researchers and practitioners agree on. Still, it is widely accepted that a community should induce a graph that is denser than the global density of the network [17]. Numerous community detection algorithms have been developed over the years, using various techniques such as optimizing modularity, removing high-betweenness edges, detecting dense subgraphs, and statistical inference. We direct the interested reader to the survey [15] or one of the numerous books on network science [22].

Most community detection algorithms aim to find a partition of the set of nodes, that is, a collection of pairwise disjoint communities with the property that each node belongs to exactly one of them. This is a natural assumption for many scenarios. For example, most of the employees on LinkedIn work for a single employer. On the other hand, users of Instagram can belong to many social groups associated with their workplace, friends, sports, etc. Researchers might be part of many research groups. A large fraction of proteins belong to several protein complexes simultaneously. As a result, many real-world networks are better modelled as a collection of overlapping communities [30] and, moreover, many community detection problems should be tackled by finding overlapping sets of nodes rather than a partition.

In the context of overlapping communities, detection is even more challenging. For example, in non-overlapping community detection one can easily verify that a node is highly connected to its proposed community, whereas in overlapping community detection a node might be assigned to many communities, and the number of connections from into each community may vary drastically.

To support the development and analysis of overlapping community detection algorithms, a large and diverse catalogue of networks with ground-truth communities is required for testing, tuning and training. Unfortunately, there are few such datasets with communities properly identified and labelled. As a result, there is a need for synthetic random graph models with community structure that resemble real-world networks to benchmark and tune unsupervised clustering algorithms. The popular **LFR** (**L**ancichinetti, **F**ortunato, **R**adicchi) model [27, 26] generates networks with communities and, at the same time, allows for heterogeneity in the distributions of both node degrees and of community sizes. Due to this structural freedom, the **LFR** model became a standard and extensively used model for generating artificial networks with ground-truth communities.

A model similar to **LFR**, the **A**rtificial **B**enchmark for **C**ommunity **D**etection (**ABCD**) [21], was recently introduced and implemented[1], along with a faster and multithreaded implementation[2] (**ABCDe**) [19]. Undirected variants of **LFR** and **ABCD** produce graphs with comparable properties, but **ABCD** (and especially **ABCDe**) is faster than **LFR** and can be easily tuned to allow the user to make a smooth transition between the two extremes: pure (disjoint) communities and random graphs with no community structure. Moreover, **ABCD** is easier to analyze theoretically—for example, in [20] various properties of the **ABCD** model are investigated, including the modularity which, despite some known issues such as the "resolution limit" reported in [16], is an important graph property in the context of community detection. In [6], some interesting and desired self-similar behaviour of the **ABCD** model is discovered; namely, that the degree distribution of ground-truth communities is asymptotically the same as the degree distribution of the whole graph (appropriately normalized based on their sizes). Finally, the building blocks in the model are flexi-

[1]https://github.com/bkamins/ABCDGraphGenerator.jl/
[2]https://github.com/tolcz/ABCDeGraphGenerator.jl/

ble and may be adjusted to satisfy different needs. Indeed, the original **ABCD** model was adjusted to include outliers (**ABCD+o**) [23] and was generalized to hypergraphs (**h–ABCD**) [24][3]. For these reasons **ABCD** is gaining recognition as a benchmark for community detection algorithms. For example, in [3] the authors use both **ABCD** and **LFR** graphs to compare 30 community detection algorithms and, in their work, highlight that *"being directly comparable to **LFR**, **ABCD** offers additional benefits, including higher scalability and better control for adjusting an analogous mixing parameter."*

In this paper we introduce the **ABCD+o**^{2} model: a generalization of the **ABCD+o** model that allows for overlapping communities. The **LFR** model has been extended in a similar way [26], and in this model the nodes are assigned to communities based on the construction of a random bipartite graph between nodes and communities resulting in (a) a small amount of overlap between almost every pair of communities, and (b) rarely any pair of communities with a large overlap. In **ABCD+o**^{2}, we instead generate overlapping communities based on a hidden, low-dimensional geometric layer which tends to yield fewer and larger overlaps. This geometric approach to community structure is justified by the fact that latent geometric spaces are believed to shape complex networks (e.g., social media networks shaped by users' opinions, education, knowledge, interests, etc.). These latent spaces have been successfully employed for many years to model and explain network properties such as self-similarity [33], homophily and aversion [18]. For more details, we direct the reader to the survey [9] or the book [32]. In addition to the geometric layer, the ancillary benefits of the **ABCD** model (an intuitive noise parameter, a fast implementation, and theoretical analysis) are still present, making the **ABCD+o**^{2} model an attractive option for benchmarking community detection algorithms.

As mentioned above, modelling complex networks using underlying geometric hidden layers is not new. However, in most spatial models such as the Spatial Preferential Attachment (SPA) model [1], nodes are embedded in a metric space, and link formation is influenced by the metric distance between nodes. The metric space is meant to be a feature space, where coordinates represent information associated with the node. For example, in text mining, documents are commonly represented as vectors in a word space. The metric is chosen so that metric distance represents similarity, i.e. nodes whose information entities are closely related will be at a short distance from each other in the metric space. In the **ABCD+o**^{2} model, the underlying geometric space plays a slightly different role. In this case, the geometric layer affects the formation of communities which group together nodes that are close to each other. For example, communities in social networks tend to group together users with similar interests, age, geographical locations, all of which can be represented as vectors in a metric space. This natural mechanism creates correlated overlapping clusters which might be challenging to obtain via other means. Connections between users in such social networks are formed as a consequence of such underlying overlapping structure. We are not aware of similar approach used in other models but, given it is quite natural, we would not be surprised if similar techniques were already used.

The remainder of the paper is organized as follows. We present the **ABCD+o**^{2} model in Section 2 with a detailed guide to its graph construction process in Section 2.5. Next, in Section 3 we show some properties of the model and compare these properties to those of real graphs with known overlapping communities. In Section 4 we demonstrate an application of the model by benchmarking community detection algorithms and comparing their quality under different levels

[3] `https://github.com/bkamins/ABCDHypergraphGenerator.jl`

of noise and overlap. Finally, some concluding remarks are given in Section 5.

2 ABCD+o^2 — ABCD with Overlapping Communities and Outliers

As mentioned in the introduction, the original **ABCD** model was extended to include outliers resulting in the **ABCD+o** model. For our current needs, we extend **ABCD+o** further to allow for non-outlier nodes to belong to multiple communities, resulting in the **ABCD+o**^{2} model, **ABCD** with **o**verlapping communities and **o**utliers.

2.1 Notation

For a given $n \in \mathbb{N} := \{1, 2, \ldots\}$, we use $[n]$ to denote the set consisting of the first n natural numbers, that is, $[n] := \{1, 2, \ldots, n\}$.

We use standard probability notation throughout the paper. For a random variable X, write $\mathbb{P}(X = k)$ for the probability that $X = k$, and write $\mathbb{E}[X]$ for the expected value of X. For a distribution $\mathcal{D}$, write $X \sim \mathcal{D}$ to mean X is sampled from the distribution $\mathcal{D}$.

For any real number $x = a + b$ with $a \in \mathbb{Z}$ and $b \in [0, 1)$, the random variable $\lfloor x \rceil$ is defined as

$$\lfloor x \rceil := \begin{cases} a & \text{with probability } 1 - b, \text{ and} \\ a + 1 & \text{with probability } b\,. \end{cases}$$

Note that $\mathbb{E}[\lfloor x \rceil] = a(1 - b) + (a + 1)b = x$.

Power-law distributions will be used to generate both the degree sequence and community sizes so let us formally define them. For given parameters $\gamma \in (0, \infty)$, $\delta, \Delta \in \mathbb{N}$ with $\delta \leq \Delta$, we define a truncated power-law distribution $\mathcal{P}(\gamma, \delta, \Delta)$ as follows. For $X \sim \mathcal{P}(\gamma, \delta, \Delta)$ and for $k \in \mathbb{N}$ with $\delta \leq k \leq \Delta$,

$$\mathbb{P}(X = k) = \frac{\int_k^{k+1} x^{-\gamma}\, dx}{\int_\delta^{\Delta+1} x^{-\gamma}\, dx}\,. \tag{1}$$

2.2 The Configuration Model

The well-known configuration model is an important ingredient of all variants of the **ABCD** models, so let us formally define it here. Suppose that our goal is to create a graph on n nodes with a given degree sequence $\mathbf{d} := (d_i, i \in [n])$, where $\mathbf{d}$ is a sequence of non-negative integers such that $m := \sum_{i \in [n]} d_i$ is even. We define a random multi-graph $\mathrm{CM}(\mathbf{d})$ with a given degree sequence known as the *configuration model* (sometimes called the *pairing model*), which was first introduced by Bollobás [10]. See [7, 36, 35] for related models and results.

We start by labelling nodes as $(v_i, i \in [n])$ and endowing each v_i with d_i half-edges. We then iteratively choose two unpaired half-edges uniformly at random (from the set of pairs of remaining half-edges) and pair them together to form an edge. This process yields a graph $G_n \sim \mathrm{CM}(\mathbf{d})$ on n nodes, where G_n is allowed self-loops and multi-edges and thus G_n is a multi-graph.

Parameter	Range	Description
n	$\mathbb{N}$	Number of nodes
s_0	$\mathbb{N}$	Number of outliers
η	$[1, \infty)$	Average number of communities a non-outlier node is part of
d	$\mathbb{N}$	Dimension of the reference layer
ρ	$[-1, 1]$	Pearson correlation between the degree of nodes and the number of communities they belong to
γ	$(2, 3)$	Exponent for power-law degree distribution
δ	$\mathbb{N}$	Min degree
Δ	$\{\delta, \delta + 1, \ldots\}$	Max degree
β	$(1, 2)$	Exponent for power-law community size distribution
s	$\{\delta + 1, \delta + 2, \ldots\}$	Min community size
S	$\{s, s + 1, \ldots\}$	Max community
ξ	$[0, 1]$	Level of noise

Table 1: Parameters of the **ABCD+o**^{2} model.

2.3 Parameters of the ABCD+o^2 Model

Table 1 summarizes the parameters that govern the **ABCD+o**^{2} model. Note that the ranges for parameters γ and β are suggestions chosen according to experimental values commonly observed in complex networks [4, 29]. Furthermore, although power-law distributions are a sensible default option, the implementation allows the user to pass a custom degree or primary community size sequence (defined in phase 1 of section 2.5).

2.4 Big Picture

The **ABCD+o**^{2} model generates a random graph on n nodes with a degree sequence and a community size sequence that by default follow truncated power laws with exponents γ and β respectively. In the first phase, we assign non-outlier nodes to communities, making use of a d-dimension hidden reference layer whose geometry is used to guide the community overlap. Next, we sample and assign degree to nodes. In phase 3, the degree of each node is split so that (on average) the proportion in the background graph is ξ and the remaining degree is split evenly among the communities to which the node belongs. We sample the background graph and each community graph using the configuration model. Finally, we perform any necessary rewiring to remove loops and multi-edges so that the final graph is simple. Figure 1 visualizes each step in an example with 15 nodes, 3 communities, 3 outliers, $\eta = 1.3$, and $\xi = 0.25$.

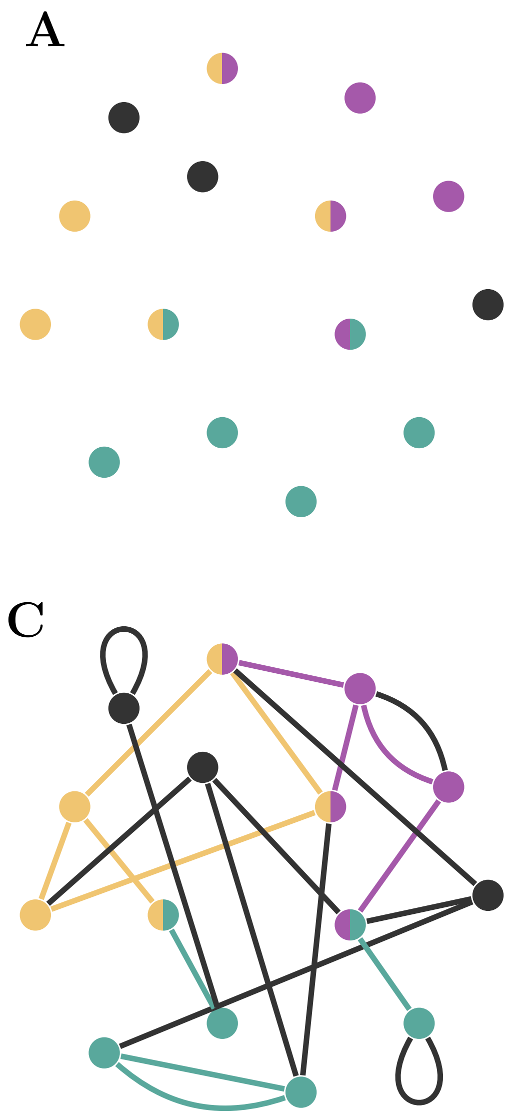

Figure 1: A big picture example of generating an **ABCD+o²** graph. In panel A, we assign nodes to communities; each colour represents a community and black nodes represent outliers. Note that some nodes belong to multiple communities. In panel B, we assign degrees to nodes and split each degree into the background degree (black stubs) and community degree (coloured stubs). In panel C, we build each graph using the configuration model. Finally in panel D, we perform edge swaps to resolve self-loops and parallel edges.

2.5 The ABCD+o^2 Construction

Phase 1: assigning communities.

This phase is the essential contribution of the **ABCD+o^2** extension. We label the nodes $v_1, \dots, v_n$ such that nodes $v_{\hat{n}+1}, \dots, v_n$ are the outliers, with $\hat{n} = n - s_0$.

The proposed method embeds nodes in a d-dimensional latent space in which we first create non-overlapping *primary* communities, and then expand each community by (in expectation) a factor of η to create overlap. The user may pass a sequence $(\hat{s}_j, j \in [L])$ of primary community sizes that we relabel so $\hat{s}_1 \geq \cdots \geq \hat{s}_L$. Otherwise, we sample the primary community size sequence from a truncated power-law $\mathcal{P}\left(\beta, \hat{s}, \hat{S}\right)$, where $\hat{s} = \lceil s/\eta \rceil$ and $\hat{S} = \lfloor S/\eta \rfloor$. We sample $\hat{s}_j \sim \mathcal{P}\left(\beta, \hat{s}, \hat{S}\right)$ independently until the sum of samples is at least $\hat{n}$. Since the sum of primary community sizes must equal $\hat{n}$, if the sum is $\hat{n} + a$ for some $a > 0$ then we fix the sequence as follows. If the last sample $\hat{s}_{\max j}$ has size at least $a + \hat{s}$, we reduce it by a. Otherwise, we delete the last sample and select $\hat{s}_{\max j} - a$ other samples to increase by 1. Let L be the random variable counting the number of communities, and relabel $(\hat{s}_j, j \in [L])$ so that $\hat{s}_1 \geq \cdots \geq \hat{s}_L$.

At this point, let us pause to ensure the intuition for η matches its use. A primary community with size $\hat{s}_j$ will expand to a final size of $s_j = \lfloor \eta \hat{s}_j \rceil$. Thus, after expanding each community, the average number of communities per non-outlier node (in expectation) is

$$\mathbb{E}\left[\frac{\sum_{j\in[L]} s_j}{\hat{n}}\right] = \frac{\sum_{j\in[L]} \mathbb{E}\left[s_j\right]}{\hat{n}} = \frac{\sum_{j\in[L]} \mathbb{E}\left[\lfloor \eta \hat{s}_j \rceil\right]}{\hat{n}} = \frac{\eta \sum_{j\in[L]} \hat{s}_j}{\hat{n}} = \frac{\eta \hat{n}}{\hat{n}} = \eta.$$

We now create the d-dimensional reference layer. One may think of this reference layer as a latent space representing node properties (such as people's age, education, geographic location, beliefs, etc.) and shaping communities (such as communities in social media) based on proximity. Each node $(v_i, i \in [\hat{n}])$ is given a vector $\mathbf{v}_i$ in $\mathbb{R}^d$ sampled independently and uniformly at random from the ball of radius 1 centered at the origin. Let $R_1 = \{\mathbf{v}_i : i \in [\hat{n}]\}$. We sequentially build communities $\hat{C}_1, \hat{C}_2, \dots, \hat{C}_L$ where, by convention, $\hat{C}_0 = \emptyset$. To create $\hat{C}_j$, we first select the vector in R_j that is furthest from the origin, and call it the seed c_j. The community $\hat{C}_j$ is set as the node associated with the seed, along with the nodes of the seed's $\hat{s}_j - 1$ nearest neighbours in R_j. Finally, we set $R_{j+1} = R_j - \{\mathbf{v}_i : v_i \in \hat{C}_j\}$. At the end of the loop, each node belongs to exactly one primary community and the collection $(\hat{C}_j, j \in [L])$ is a partition of the nodes.

Next, we grow each primary community $\hat{C}_j$ of size $\hat{s}_j$ to the full community C_j of size s_j. Each community will grow independently, so we may grow them in any order (or in parallel). Let $\mathbf{x}_j \in \mathbb{R}^d$ be the center of mass (mean) of $\{\mathbf{v}_i : v_i \in \hat{C}_j\}$. The full community C_j is then the union of $\hat{C}_j$, and the $s_j - \hat{s}_j$ nodes not in $\hat{C}_j$ with vectors closest to $\mathbf{x_j}$. In effect, we have expanded $\hat{C}_j$ around it's center of mass. We call the nodes in $C_j - \hat{C}_j$ the *secondary* members of community j, and likewise for node $v \in C_j - \hat{C}_j$ we call C_j a *secondary* community of v.

We now have each $v_1, \dots v_{\hat{n}}$ assigned to exactly one primary community and possibly some secondary communities. In Figure 2 we show an example of the reference layer with $\hat{n} = 150$ and show the construction of C_3, with $\hat{s}_3 = 40$ and $\eta = 1.75$. From a computational perspective, to ensure that finding primary and secondary communities is efficient, we use k-d trees to perform spatial lookups [8].

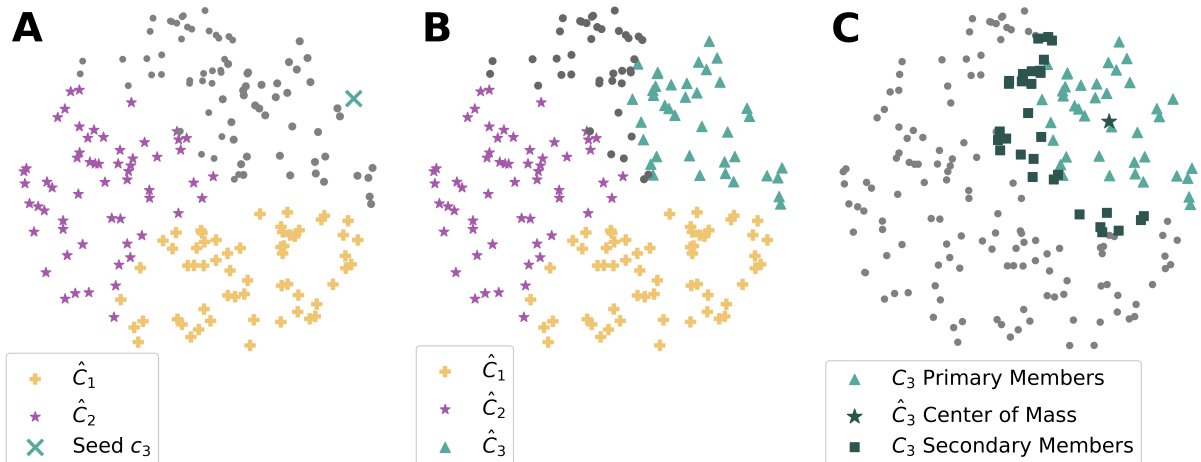

Figure 2: Example of building the community C_3 with $\hat{s}_3 = 40$ and $\eta = 1.75$ using a 2-dimensional reference layer. First, in panel A, we select the seed c_3 as the vector furthest from the origin that does not have a primary community. Next (panel B), we build the primary community $\hat{C}_3$ as node of the seed and the nodes of the seed's $\hat{s}_3 - 1$ nearest neighbours that do not yet have a primary community. Finally, in panel C, we expand create C_3 by expanding $\hat{C}_3$ by a factor of η (from 40 to $40 \cdot 1.75 = 70$ nodes) by taking the nearest neighbours to the primary center of mass.

Phase 2: assigning degrees.

By default, we sample the degree sequence $(d_i, i \in [n])$ independently from $\mathcal{P}(\gamma, \delta, \Delta)$. Since the total degree in a simple graph must be even, if $\sum_{i \in [n]} d_i$ is odd, we decrease a maximal d_i by 1. Alternatively, the degree sequence can be given explicitly as an input. Finally, we relabel the sequence so that $d_1 \geq d_2 \geq \ldots, \geq d_n$.

The goal of outliers is to sample their neighbours from the entire graph, ignoring the underlying community structure. However, a problem may occur if ξ is close to zero. In the extreme case, where $\xi = 0$, the edges in the background graph are precisely the edges connecting two outliers, and for a simple background graph to exist all outliers must have degree at most $s_0 - 1$. To handle the more general issue that arises when ξ is small, we estimate the number of nodes with at least one edge in the background graph to upper-bound the allowable degree of an outlier. If we imagine the degrees were assigned at random, we would expect

$$\ell = (\hat{n}/n) \sum_{i \in [n]} \min(1, \xi d_i)$$

non-outlier nodes with positive background degree (a non-outlier with degree d will have background degree $\lfloor \xi d \rceil$, see Phase 3). Moreover, since the s_0 outliers will assign their entire degree to the background graph, in total we expect $\ell + s_0$ nodes with positive background degree. Therefore, like in the **ABCD+o** model [23], we insist that a degree d cannot be assigned to an outlier unless

$$d \leq \ell + s_0 - 1. \tag{2}$$

In practice, the number of nodes n is large, the number of outliers s_0 is relatively small, and the level of noise ξ is not zero, so there are plenty of nodes with a non-zero degree in the background graph, and so there is no restriction needed for outliers. A subset of degrees satisfying (2) is selected uniformly at random and assigned to the outliers.

Let $(\hat{d}_i, i \in \hat{n})$ be the subsequence of degrees not assigned to outliers. Similar to previous paragraph, to build simple community graphs we must avoid large degrees being assigned to nodes in small communities. Let $\eta_i = |\{j : v_i \in C_j\}|$ be the number of communities containing v_i. When creating edges in Phase 3, node v_i with degree d will have at most degree $\left\lceil \lceil (1-\xi)d \rceil / \eta_i \right\rceil$ assigned to each of its communities (we split the community degree as evenly as possible between communities). So, if v_i is assigned degree d, then for each community C_j containing v_i, $|C_j|$ must be at least $\left\lceil \lceil (1-\xi)d \rceil / \eta_i \right\rceil + 1$ to accommodate the community neighbours of v_i. This is a lower bound, as additional neighbours of v_i coming from the background graph might end up in C_j as well. To account for the extra background neighbours, the **ABCD+o** model [21] uses a small correction guided an internal parameter ϕ. Extending the derivation of ϕ to overlapping communities would require analysis of each overlap, which would be computationally expensive. Instead, we keep the same equation as **ABCD+o**, and let

$$\phi = 1 - \sum_{j \in [L]} \left(\frac{\hat{s}_j}{\hat{n}} \right)^2 \frac{\hat{n}\xi}{\hat{n}\xi + s_0} .$$

Typically, ϕ is very close to 1 and so this correction tends to be negligible in both theory and in practice, but for consistency we keep ϕ in the **ABCD+o^2** model.

We can now describe the pairing process of unassigned degrees $(\hat{d}_i, i \in [\hat{n}])$ to non-outlier nodes $(v_k, k \in [\hat{n}])$. To guide the correlation between degree and number of communities, we introduce an internal parameter $\alpha \in \mathbb{R}$ that we discuss in the following paragraph. Recall that $(\hat{d}_i, i \in [\hat{n}])$ is monotone decreasing. We assign degrees sequentially, starting with $\hat{d}_1$. To assign $\hat{d}_i$, let U_i be the collection of nodes without an assigned degree. We assign $\hat{d}_i$ to node v, chosen from the subset of U_i satisfying

$$\hat{d}_i \leq \frac{\eta_k}{1 - \xi\phi} \cdot \min \left\{ |C_j| - 1 : v_k \in C_j \right\}, \tag{3}$$

with probability proportional to η_k^α. If no nodes in U_i satisfy (3), we sample v from the subset of U_i for which the right side of (3) is maximal.

After a pairing, we compute the obtained correlation ρ_α and determine if α should be larger ($\rho_\alpha < \rho$) or smaller ($\rho_\alpha > \rho$). We find an appropriate value for α with a binary search, starting with $\alpha_{\min} = -60$ and $\alpha_{\max} = 60$ and stopping once the desired precision is achieved, or after the obtained correlation level stops improving. In some cases it is impossible to reach the correlation level required by the user, in which case the closest possible correlation is produced. The relationship (beyond monotone) between ρ and α is unclear, but we feel it is easier for a user to reason about a desired ρ.

Phase 3: generating graphs.

First, we split each degree into the community degrees and the background degree. For each non-outlier node v_i with degree d_i, define random variables $Y_i = \lfloor (1-\xi)d_i \rceil$ and $Z_i = d_i - Y_i$ (note that $\mathbb{E}[Y_i] = (1-\xi)d_i$ and $\mathbb{E}[Z_i] = \xi d_i$) and split the degree d_i into Y_i community degree and Z_i

background degree. The community degree is further split as evenly as possible into the η_i communities of v_i. Specifically, when splitting the community degree of v_i, $Y_i - \eta_i \lfloor Y_i/\eta_i \rfloor$ communities (chosen randomly) each receive degree $\lceil Y_i/\eta_i \rceil$ and the remaining communities each receive degree $\lfloor Y_i/\eta_i \rfloor$. Recall that by (3), we have already ensured that each community containing v_i has at least $\lceil Y_i/\eta_i \rceil + 1$ nodes. For the outliers, the entire degree is assigned to the background.

For each $j \in [L]$, we independently construct the *community graph* $G_{n,j}$ as follows. From the previous paragraph, each $v_i \in C_j$ has some degree allocated to community C_j, and form the community graph $G_{n,j}$ by passing these degrees to the configuration model (Subsection 2.2). In the event that the sum of degrees in a community is odd, we pick a node with maximum degree in this community, decrease the community degree by one, and add one to the background degree. Finally, we construct the *background graph* $G_{n,0}$ as per the configuration model on node set $[n]$ and degree sequence $(Z_i, i \in [n])$.

Phase 4: rewiring self-loops and multi-edges.

Note that, although we are calling $G_{n,0}, G_{n,1}, \ldots, G_{n,L}$ *graphs*, they are in fact *multi-graphs*. To ensure that G_n is simple, we perform a series of *rewirings*. A rewiring takes two edges as input, splits them into four half-edges, and creates two new edges distinct from the input. We first rewire each community graph $(G_{n,j}, j \in [L])$, and the background graph $G_{n,0}$, independently as follows.

1. For each edge $e \in E(G_{n,j})$ that is a loop, we add e to a *recycle* list that is assigned to $G_{n,j}$. Similarly, if $e \in E(G_{n,j})$ contributes to a multi-edge, we add all but one copy of this edge to the *recycle* list.

2. We shuffle the *recycle* list and, for each edge e in the list, we choose another edge e' uniformly from $E(G_{n,j}) \setminus \{e\}$ (not necessarily in the list) and attempt to rewire these two edges. If the rewiring does not lead to any further self-loops or multi-edges, we save the result and remove e (and e' if present) from the recycle list.

3. If the recycle list is smaller than before the previous step, we repeat step 2. Otherwise, we give up and move all the "bad" edges from the community graph to a *global recycle* list.

As a result, ignoring edges in the *global recycle* list, all community graphs (including the background graph) are simple. However, as is the case in the original **ABCD** model, an edge in the background graph can form a multi-edge with an edge in a community graph. Another problem, specific to **ABCD+o**^{2}, is that an edge from one community can form a multi-edge with an edge from a different but overlapping community. Both types of problematic edges are added to the *global recycle* list. We merge all community graphs with the background graph, and rewire the *global recycle* list following the same process as before (steps 2 and 3 above with $G_{n,j}$ replaced by the full graph). Since the graph is typically sparse, this final rewiring is usually very fast. However, in rare cases it is possible that the process does not terminate. We set a limit to attempt rewiring at most 100 times the size of the global recycle list. If, after this many iterations, we cannot find any new admissible edge (thus reducing the size of global recycle list) we throw an error and indicate to the user that the algorithm failed to terminate with success for the given input parameters (indicating that the likely cause is that the required input degree sequence was not graphic, and should be reconsidered by the user).

3 Properties of the ABCD+o^2 Model

In this section, we present experiments highlighting the properties of the **ABCD+o^2** model. We begin by investigating the communities generated by the reference layer as described in Section 3.1. We then continue with additional features of the model in Section 3.2. We are mainly interested in comparing the model's properties with properties of real-world networks. Thus, for the model parameters, we use values derived from several networks that have known overlapping communities. The networks were presented in [37], and are available on the Stanford Network Analysis Project (SNAP) website[4].

DBLP: a network where nodes are authors and an edge exists between two authors if they published a paper together. The ground-truth communities are defined by the publication venues of the journals in which the authors published.

Amazon: a network where nodes are products and an edge exists between two products if they are frequently co-purchased. The ground-truth communities are defined by product categories.

YouTube: a network where nodes are YouTube channels and an edge exists between two channels if they are friends on the platform. The ground-truth communities are user-generated social groups. In the full network over 95% of nodes are outliers and the level of noise ξ is 0.96. We consider the subgraph induced by the non-outlier nodes to avoid such extreme values.

For each network, all parameters other than δ and s are measured directly from the data. We set $\delta = 5$ to avoid cases where nodes have no neighbours in their own community, and insist on $s \geq 10$. The power-law exponents γ and β are fit using maximum-likelihood estimation described in [12] and implemented in the Python package **powerlaw**[5][2]. In this fitting process, we set the minimum degree as $\delta = 5$ and the minimum community size as $s = 10$ for consistency. The level of noise ξ is computed directly from the graphs as the fraction $|E_b|/|E|$ where E_b is the set of edges where the incident nodes share no communities. The parameters for each network are shown in Table 2.

Additionally, most of the experiments are affected by the dimension d of the hidden reference layer. In such cases, we compare dimensions 2, 8, and 64.

3.1 Effects of the Reference Layer

The new and novel feature of the **ABCD+o^2** model is the hidden reference layer used to generate overlapping communities. Here, we analyze the effects of this reference layer on the community overlap sizes. For comparison, we consider the **CKB** model [11] following a description from [31]. The **CKB** model is an application of the random bipartite community affiliation graph proposed for overlapping **LFR** [26] that uses a power-law distribution for the number of communities per node. Let us briefly summarize this graph construction process. The community affiliation graph is a bipartite graph where one part consists of nodes and the other consists of communities. As input, the **CKB** model requires the number of nodes n, a truncated power-law distribution $\mathcal{P}(\Omega, x_{\min}, x_{\max})$ for the number of communities per node, and another truncated power-law distribution $\mathcal{P}(\beta, s, S)$

[4] `https://snap.stanford.edu/data/index.html#communities`

[5] `https://github.com/jeffalstott/powerlaw`

Parameter	Description	DBLP	Amazon	YouTube
n	Number of nodes	317,080	334,863	52,675
s_0	Number of outliers	56,082	17,669	0
η	Average number of communities a non-outlier node is part of	2.76	7.16	2.45
ρ	Pearson correlation between the degree of nodes and the number of communities they belong to	0.76	0.22	0.37
γ	Exponent of power-law degree distribution	2.30	3.04	1.87
δ	Min degree	5	5	5
Δ	Max degree	343	549	1,928
β	Exponent of power-law community size distribution	1.88	2.03	2.13
s	Min community size	10	10	10
S	Max community size	7,556	53,551	3,001
ξ	Level of noise	0.11	0.11	0.59

Table 2: The parameters used for generating **ABCD+o**^{2} graphs based on the DBLP, Amazon, and YouTube datasets.

for the community sizes. The number of communities is set to

$$\left\lfloor \frac{n \cdot \mathbb{E}[\mathcal{P}\left(\Omega, x_{\min}, x_{\max}\right)]}{\mathbb{E}[\mathcal{P}\left(\beta, s, S\right)]} \right\rfloor .$$

First, each node in the bipartite graph is assigned a number of half-edges sampled from $\mathcal{P}\left(\Omega, x_{\min}, x_{\max}\right)$, and likewise each community is assigned a number of half-edges sampled from $\mathcal{P}\left(\beta, s, S\right)$. A small adjustment is made to ensure that the number of half-edges in both parts is equal. Lastly, the nodes are matched with the communities based on the bipartite configuration model. Thus, each community is populated with nodes, and each node is assigned to some number of communities.

Community sizes.

In the first experiment (see Figure 3), we compare the distributions of community sizes. Of course, since both **ABCD+o**^{2} and **CKB** sample community sizes directly from the specified distribution, it is unsurprising that the empirical distributions are near-perfect fits when accounting for the imposed minimum community size of 10. Thus, this first experiment merely acts as a sanity check ensuring that the **ABCD+o**^{2} and **CKB** models generate community sizes correctly. Moreover, this experiment verifies that the distributions coming from the real networks are indeed power-law.

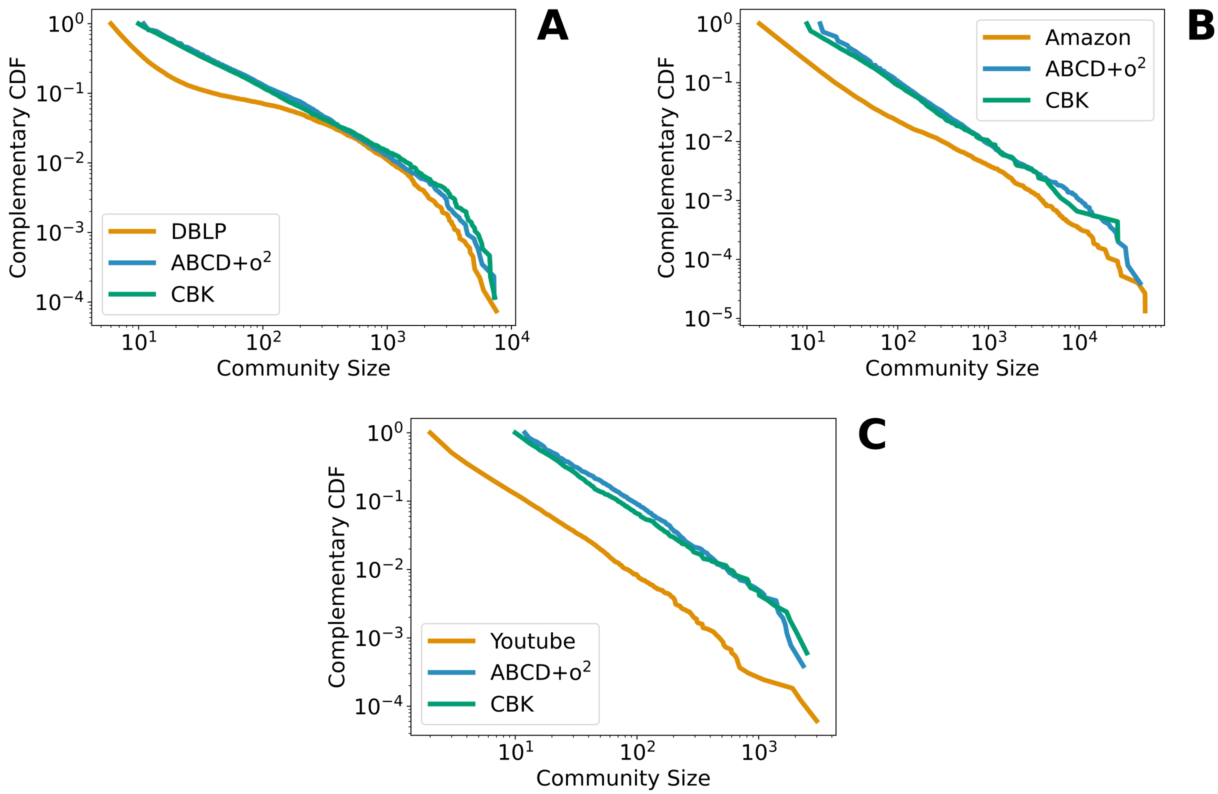

Figure 3: Distribution of community sizes for real-world networks and their synthetic counterparts: DBLP in panel A, Amazon in panel B, and Youtube in panel C. The y-axis shows the empirical complementary cumulative distribution function, computed as $y = 1 - |\{C_i : |C_i| < x\}| \; / \; |\{C_i\}|$.

Number of communities per node.

Results are presented in Figure 4. In contrast with the previous experiment, there is less agreement between the two models and the real distributions. In the case of the YouTube and DBLP networks, the power-law distribution generated by the **CKB** model is a good fit for the empirical distribution, whereas the Amazon distribution does not appear to be power-law. In contrast, the **ABCD+o**^{2} model is able to produce a distribution similar to each of the three networks with a properly tuned dimension parameter d; for the Amazon graph a low dimensional reference layer captures the behaviour well, and for the DBLP and YouTube graphs a high dimensional reference layer captures the behaviour well. The flexibility and realistic distributions are even more encouraging when we recall that the distribution is not given to the **ABCD+o**^{2}model, and is instead a natural feature of the hidden reference layer.

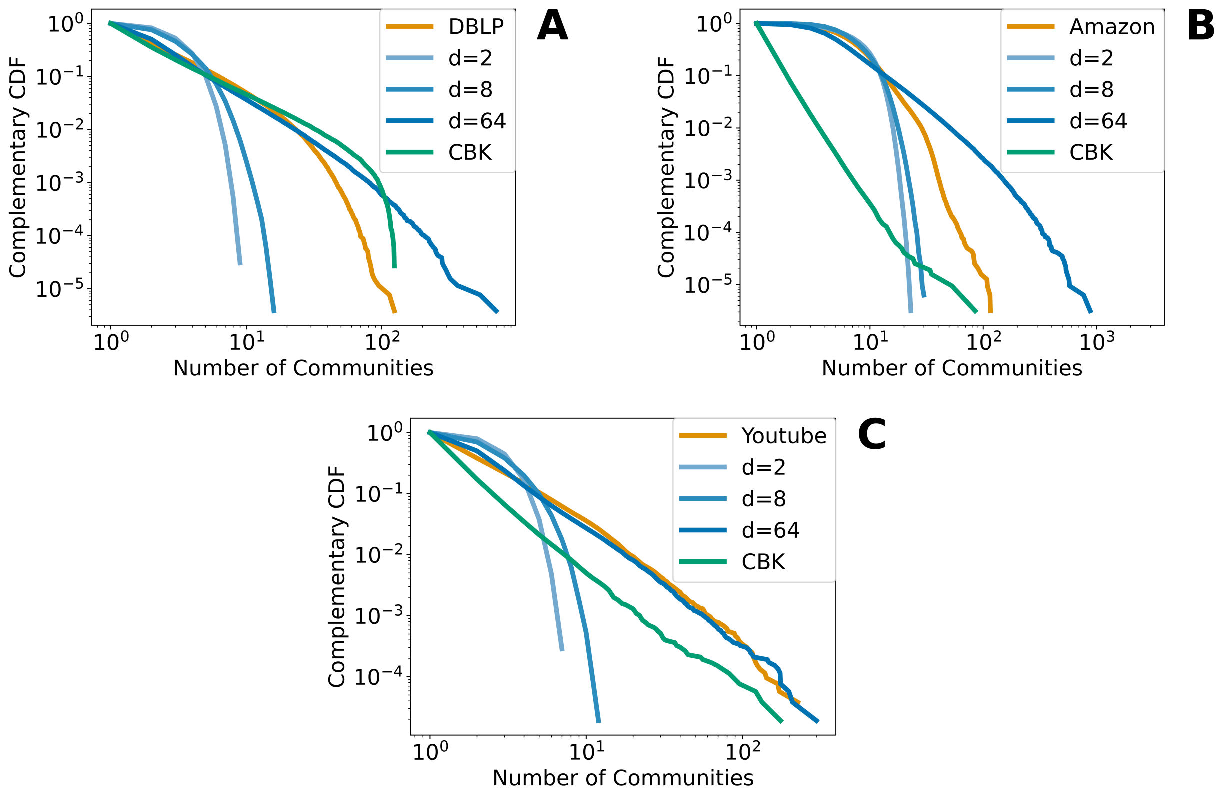

Figure 4: Distribution of the number of communities per node in each real-world network and their synthetic counterparts: DBLP in panel A, Amazon in panel B, and Youtube in panel C. The y-axis shows the empirical complementary cumulative distribution function, computed as $y = 1 - |\{v_i : |\{C_j : v_i \in C_j\} < x\}| \ / \ |V|$.

Community intersection sizes.

Results are presented in Figure 5. We examine the size of the overlaps produced by each model. Neither the **ABCD+o**^{2} model nor the **CKB** model specifies a particular distribution, but we can see the **ABCD+o**^{2} model, with a well-tuned dimension parameter, is able to create a similar distribution to the real graph. In contrast, the **CKB** model does not fit the Amazon and YouTube graphs well. Specifically, the **CKB** model produces too many small intersections and, moreover, the largest intersection is several orders of magnitude too small. We consider this experiment strong evidence for using a reference layer as the default option for generating overlapping communities.

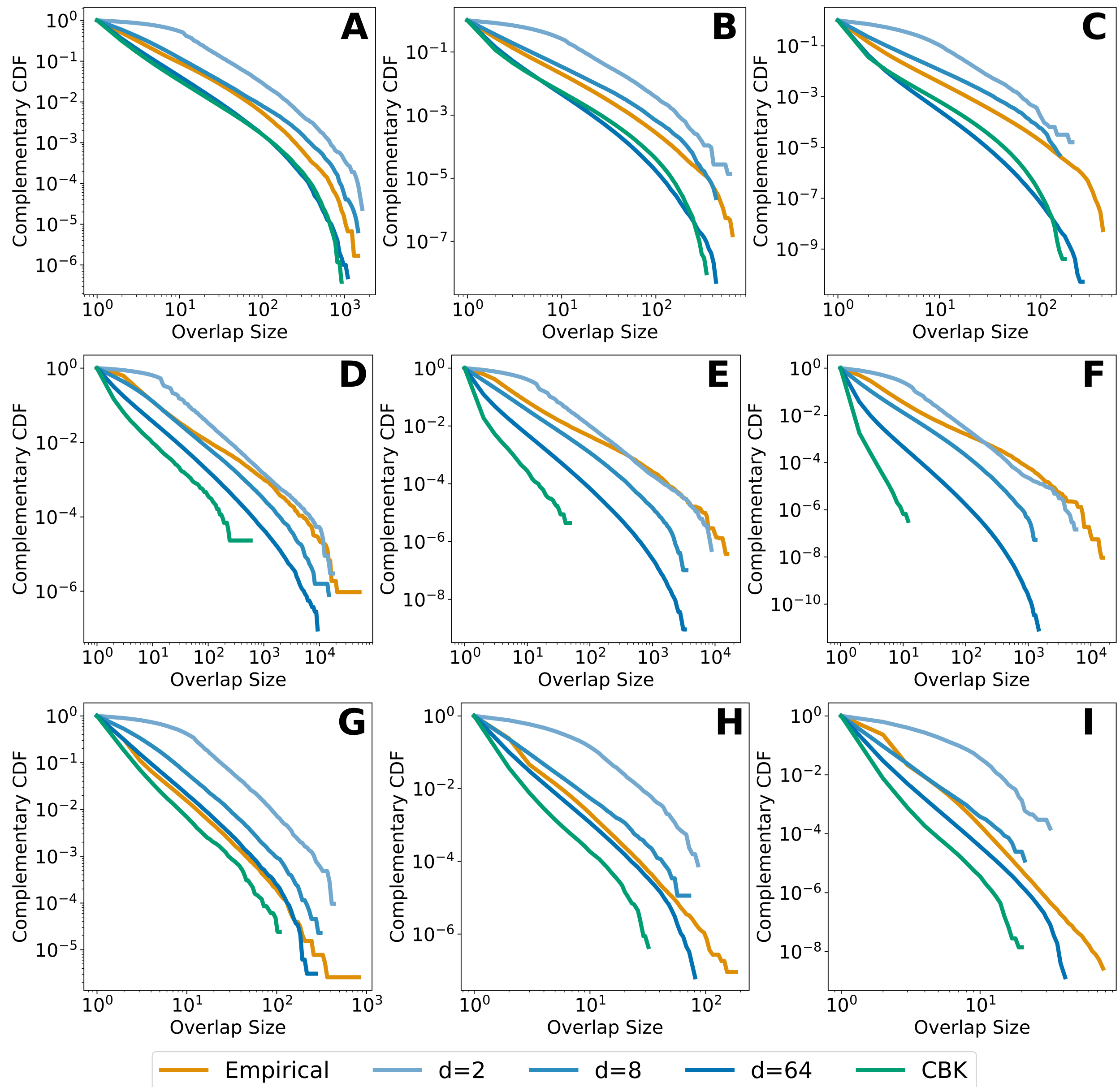

Figure 5: Distribution of the size of overlaps in real graphs and their synthetic counterparts. Each row corresponds to a real graph, panels A,B,C for DBLP, panels D,E,F for Amazon, and panels G,H,I for Youtube. The left (panels A,D,G), center (panels B,E,H) and right (panels C,F,I) columns correspond to overlaps of two, three, and four communities respectively. The y-axis shows the empirical complementary cumulative distribution function of non-empty overlaps. For example, the center column of figures corresponds to three overlapping communities and is computed as $y = 1 - |\{\{C_i, C_j, C_k\} : |C_i \cap C_j \cap C_k| < x\}| \,/\, |\{\{C_i, C_j, C_k\} : |C_i \cap C_j \cap C_k| \neq \emptyset|$.

Graph	Empirical	$d = 2$	$d = 8$	$d = 64$
DBLP	0.76	0.42	0.56	0.68
Amazon	0.22	0.20	0.19	0.20
YouTube	0.37	0.37	0.36	0.38

Table 3: Measured ρ in both the real networks and the corresponding **ABCD+o**^{2} graphs. We generated 10 **ABCD+o**^{2} graphs for each dimension d of the reference layer, and found that across all graphs and dimensions, the standard error, to two decimal places, was 0.

3.2 Additional properties of the model.

We now turn to properties of the **ABCD+o**^{2} model beyond those governed by the hidden reference layer.

Degree vs. number of communities.

The **ABCD+o**^{2} model attempts to produce a correlation ρ between the degree of a node and the number of communities it belongs to. In Table 3 we report the desired correlation ρ for each of the networks, as well as the achieved correlation for the **ABCD+o**^{2} graphs. Note that the extreme positive correlation found in the DBLP network cannot be matched by the model, although it can still achieve a strong positive correlation. The maximum achievable correlation increases with the dimension as there are more nodes belonging to a large number of communities (see Figure 4).

Density of intersections

One consequence of forcing nodes of large degree into community intersections is that those intersections will be denser than the individual communities. We show this phenomenon in Figure 6 by comparing the community and intersection densities for various ρ while fixing the other parameters. Note that, given communities C_j and C_k with $s_j \gg s_k$, the density of C_k is almost always larger than that of C_j since both community graphs are sparse (at most a $1 - \xi$ fraction of each node's degree is delegated to one of its communities). Figures for the other graphs and dimensions can be found in Appendix A.

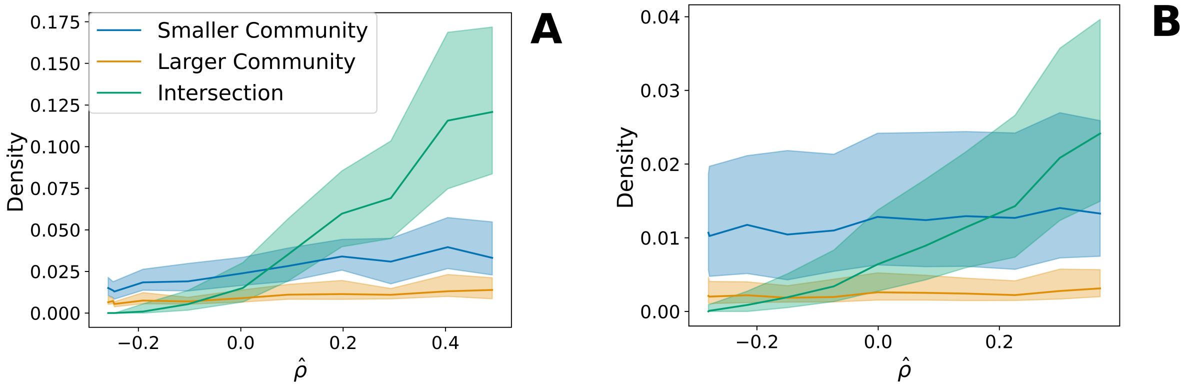

Figure 6: Community overlap densities compared to individual community densities for various values of ρ in Youtube-like graphs (panel A) and DBLP-like graphs (panel B) with $d = 8$. For 9 **ABCD+o**^{2} graphs with ρ evenly spaced in $[-0.5, 0.5]$, we compute the empirical value $\hat{\rho}$ and the distribution of density for overlapping communities. We plot $\hat{\rho}$ against the distribution of the density for the intersection and each community individually. The line corresponds to the median, and the shaded region to the 25th to 75th quantile. For computational reasons, we only consider overlaps of size at least 25, and to ensure we are not simply measuring the density of the smaller community, the size of the overlap must be at most half the size of the smaller community.

Having community intersections that are denser than the respective communities is a desirable property and one of our main motivators for introducing the parameter ρ to the **ABCD+o**^{2} model. In [38] it was found that, in real networks, community intersections can be significantly denser than the individual communities. In Figure 7 we confirm that the DBLP, Amazon, and YouTube networks all exhibit this phenomenon. Moreover, we compare these densities with those of the corresponding **ABCD+o**^{2} graphs. Aside from confirming that the model can produce behaviour similar to the real networks, we also find that the dimension of the reference layer affects the density of both the communities and the intersections. Moreover, we see that a low-dimensional reference layer yields a better approximation of the real data in the case of Amazon, whereas a high-dimensional reference layer yields a better approximation in the cases of DBLP and YouTube. In the case of DBLP, the difference in distribution between dimensions is partially attributable to the difference in the empirical ρ (recall with Table 3 that the low dimension graphs could not achieve a large correlation. Note that the most similar dimension for each real graph is consistent with the experiment presented in Figure 4.

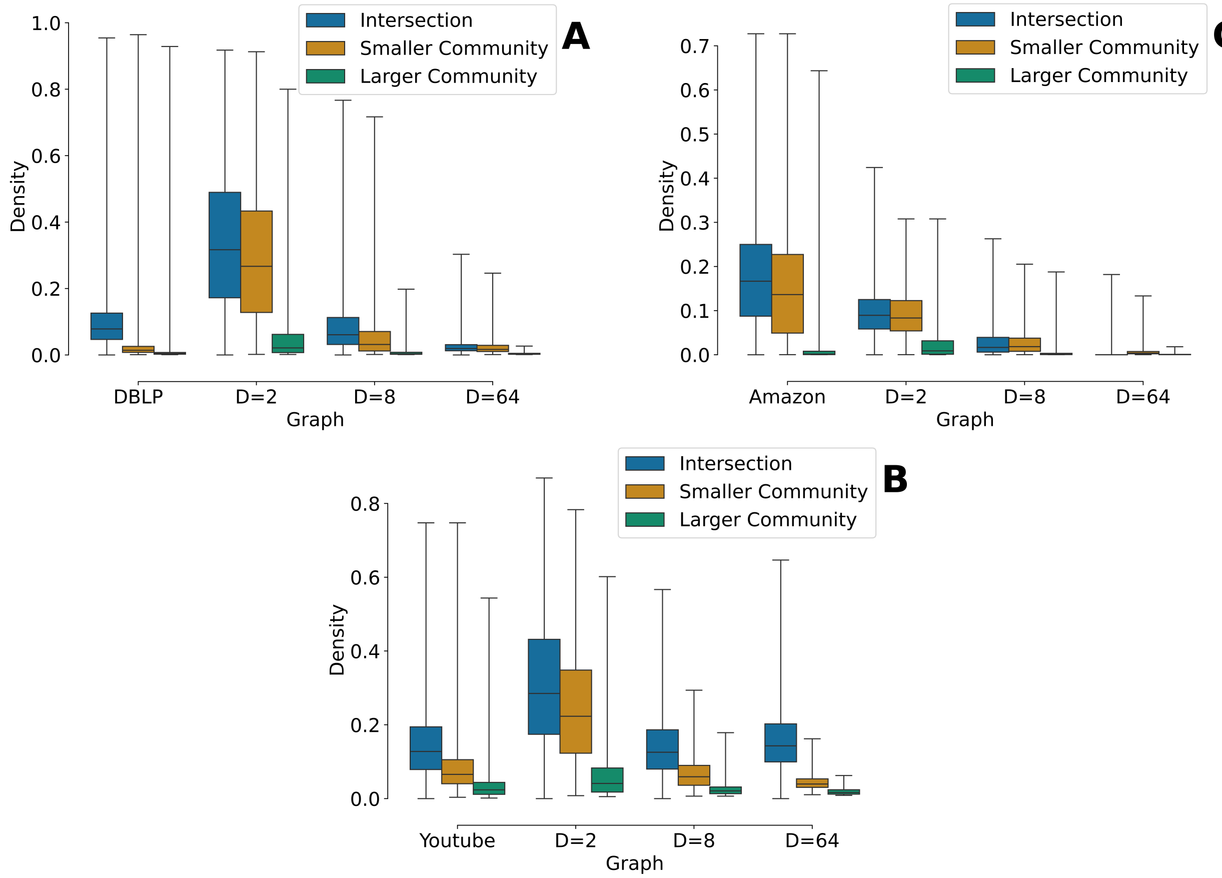

Figure 7: Distribution of the density for pairs of communities with non-zero overlap in real graphs and their synthetic counterparts; DBLP in panel A, Amazon in panel B, and Youtube in panel C. The shaded region corresponds to the 25th to 75th quartiles, and the whiskers correspond to the maximum and minimum values. We restrict to overlaps of size at least 10 due to the discrete and unpredictable nature of small communities.

Internal edge fraction

Finally, we analyze how strongly nodes are connected to their communities. There are many measures for community association strength, some of which we recently analyzed in [5]. For the sake of simplicity, the measure we will use here is the internal edge fraction $\mathrm{IEF}(v, C)$, defined as

$$\mathrm{IEF}(v, C) = \frac{|\{\{u, v\} \in E : u \in C\}|}{\deg(v)}\,.$$

For a graph G, a collection of communities $\{C_i, i \in [k]\}$, and a node v, let $C_1^{(v)}, \ldots, C_k^{(v)}$ be an ordering of the communities such that

$$\mathrm{IEF}(v, C_1^{(v)}) \geq \mathrm{IEF}(v, C_2^{(v)}) \cdots \geq \mathrm{IEF}(v, C_k^{(v)})\,.$$

Then, if v is a member of ℓ communities, we would expect a jump in value between $\mathrm{IEF}(v, C_\ell^{(v)})$ and $\mathrm{IEF}(v, C_{\ell+1}^{(v)})$. In Figure 8 we show the range of the top five internal edge fractions, sorted in descending order, on DBLP (left) and a generated **ABCD+o^2** graph (right), binned by the number of communities the node v is in. As predicted, a node in ℓ communities tends to have ℓ relatively large internal edge fractions. Furthermore, despite the **ABCD+o^2** model splitting the community degree of a node v evenly among its ℓ communities, it is possible for v to have a higher internal edge fraction into community C if either some of v's edges lies in an intersection containing C or some of v's background edges land in C. Thus, we see a pattern in **ABCD+o^2** graphs where there are ℓ large but unequal internal edge fraction values. This feature is also present in the DBLP graph, the other real graphs, and the other **ABCD+o^2** graphs (see Appendix B for a complete collection of figures).

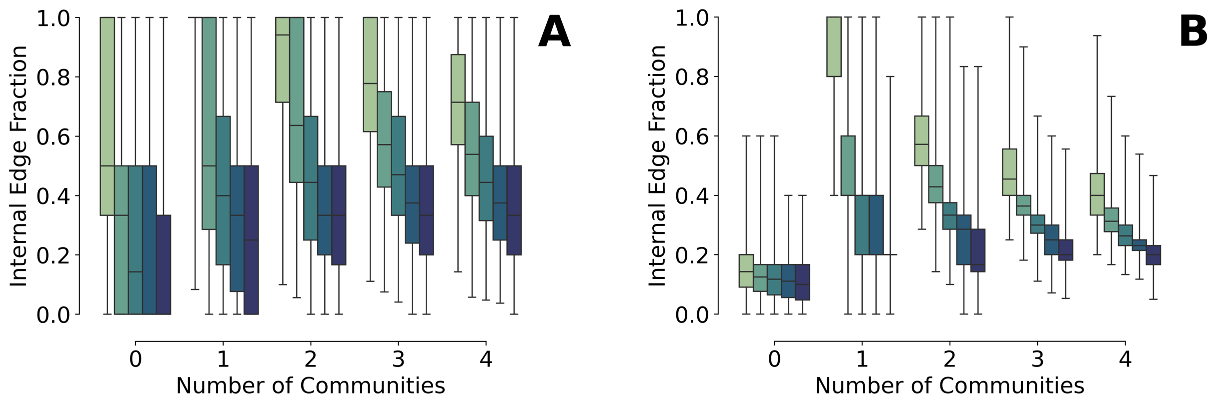

Figure 8: Five largest internal edge fractions, sorted in descending order, grouped by the number of communities. Panel A corresponds to the DBLP graphs, and panel B to the synthetic counterpart with reference layer dimension $d = 8$. A pattern is consistent between DBLP and **ABCD+o^2** , namely, that for nodes belonging to ℓ communities there are ℓ large, but not equal, internal edge fraction values.

4 Benchmarking Community Detection Algorithms

The main purpose of having synthetic models with ground-truth community structure is to test, tune, and benchmark community detection algorithms. To showcase **ABCD+o^2** in this light, we use the model to evaluate the performance of four community detection algorithms. The algorithms are as follows.

Leiden [34]: a greedy algorithm that attempts to optimize the modularity function. Note that this algorithm returns a partition, and we use it merely as a baseline to compare with algorithms that attempt to find overlapping communities.

Edge Clustering [25]: an edge-partitioning algorithm that translates to overlapping clusters of nodes. Pairs of edges are measured based on similarity of neighbourhoods, and these similarity measures dictate the order in which edge-communities merge. As edge-communities merge, the modularity is tracked on the line-graph, and the maximum modularity attained yields the edge-communities, which in turn yields overlapping node-communities.

Ego-Split [14]: a method which finds overlapping clusters in a graph G by applying a partitioning algorithm such as Leiden to an auxiliary graph G' and mapping the resulting partition onto G. The auxiliary graph G' is constructed from G by creating multiple copies, or "egos", of each node based on its neighbourhood.

Ego-Split+IEF [5]: the same algorithm as Ego-Split, but with a post-processing step that reassigns nodes to communities based on the IEF measure.

This is by no means an exhaustive list of community detection algorithms. Moreover, we use our own straightforward Python implementations of the above algorithms which may not be optimal. We wish only to showcase the usefulness of **ABCD+o**^{2} in comparing various algorithms related to the community structure of graphs.

In the first experiment, we compare the accuracy of each algorithm with respect to the ground-truth communities. The measure we use to determine accuracy is the overlapping Normalized Mutual Information (oNMI) measure [28]. For this experiment we fix the number of nodes $n = 5,000$, the minimum degree $\delta = 5$, the maximum degree $\Delta = 100$, the minimum community size $s = 50$, and the maximum community size $S = 500$. The parameters of **ABCD+o**^{2} with the most influence on the quality of detection algorithms are ξ (the level of noise) and η (the average number of communities a non-outlier is part of), and we consider every combination of $\xi \in \{0.1, 0.2, \ldots, 0.6\}$ and $\eta \in \{1, 1.25, \ldots, 2.5\}$. We fix the remaining parameters according to the YouTube graph, namely, $\gamma = 1.87$, $\beta = 2.13$, and $\rho = 0.37$.

Figure 9 summarizes the results of the experiment. We see that the **Leiden** algorithm performs the best when $\eta = 1$, whereas for $\eta > 1$ the **Ego-Split+IEF** algorithm performs better. Given that the **Leiden** algorithm finds a partition of the nodes, it is unsurprising that it quickly depreciates in quality as η increases. We also see a general trend of all algorithms performing worse as the graph gets noisier, either by increasing ξ or η. This trend is highlighted separately in Figure 10 where we consider only results with $\eta = 1.25$ (left) or $\xi = 0.1$ (right). From numerous and varying tests, we have found that, in general, increasing η is far more damming to community detection algorithms than increasing ξ.

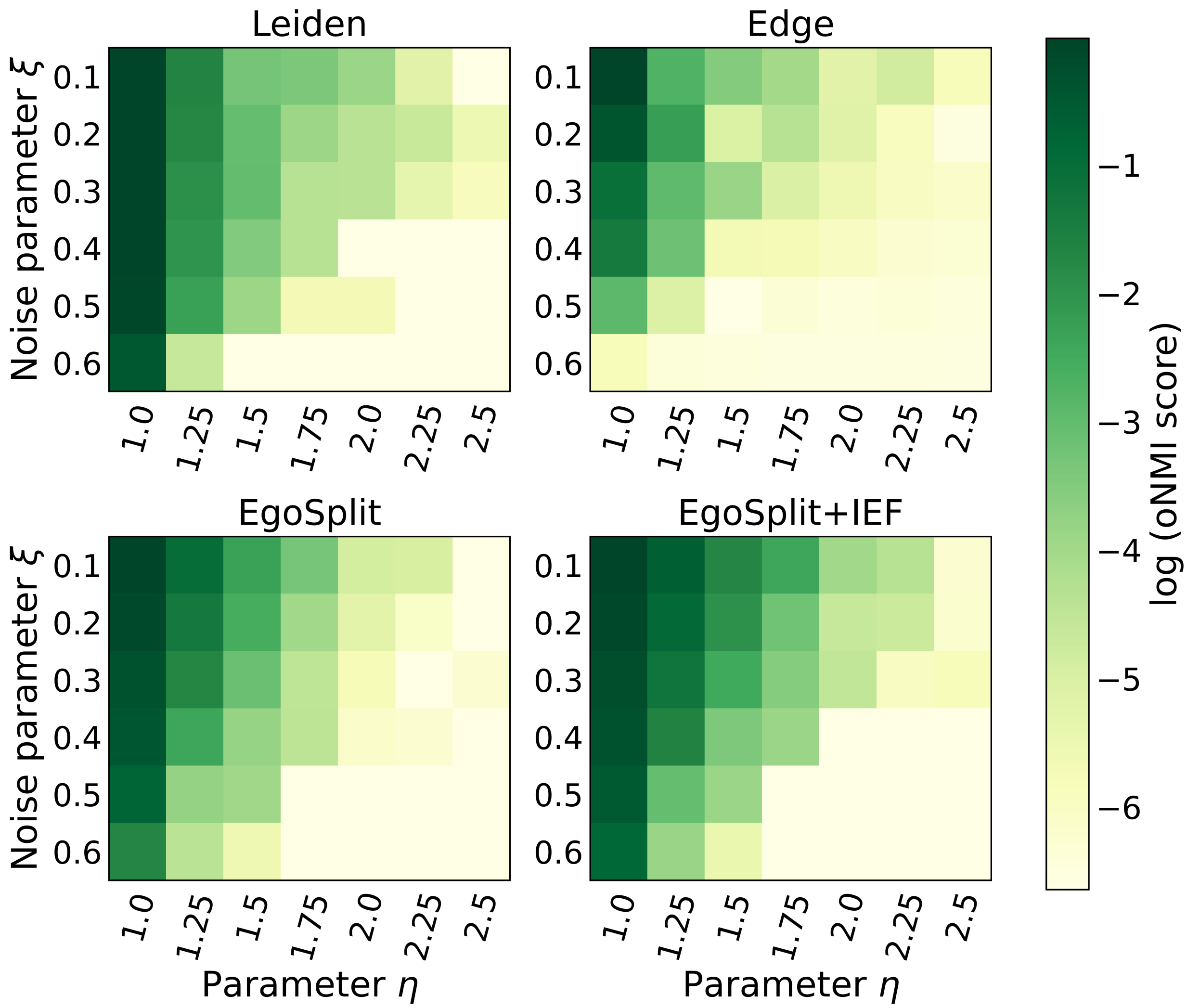

Figure 9: Comparing oNMI of four clustering algorithms on $\mathbf{ABCD{+}o^2}$ graphs with 5,000 nodes.

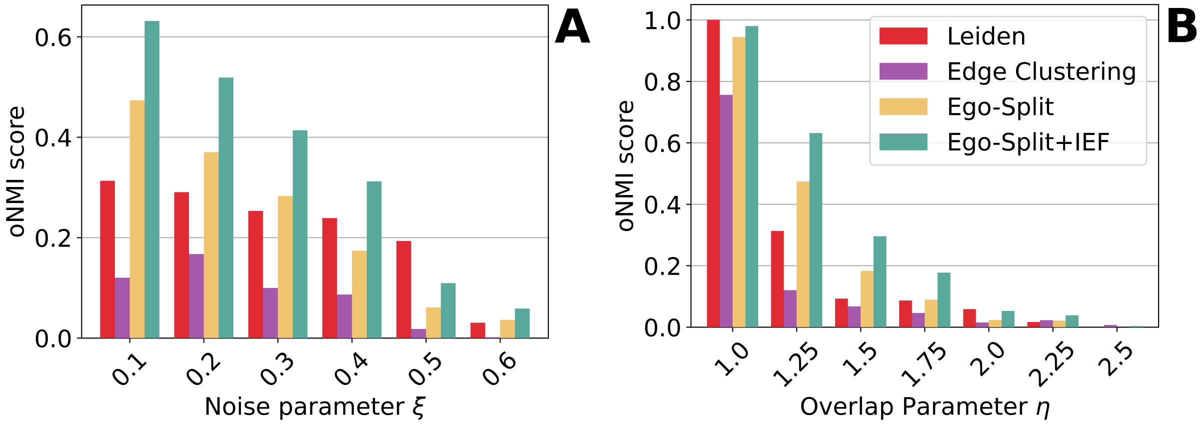

Figure 10: Comparing oNMI of four clustering algorithms on **ABCD+o**^{2} graphs with 5,000 nodes. In the panel A, we fix $\eta = 1.25$ and vary ξ, whereas in panel B, we fix $\xi = 0.1$ and vary η.

Another important aspect of clustering algorithms is their performance when running on large graphs. In our second experiment, we fix $\eta = 1.25$ and $\xi = 0.1$ and we vary the number of nodes from $n = 1,000$ to $n = 15,000$; the parameters other than n, η and ξ are the same as in the previous experiment. The results are presented in Figure 11. We see that all algorithms trying to recover overlapping communities are significantly slower than the simple **Leiden** partitioning algorithm. We also see that adding the IEF-based post-processing stage to the Ego-Splitting algorithm is negligible in terms of performance. Together with the experiment presented in Figure 9, this negligible slowdown suggests that including a post-processing stage to an overlapping community detection algorithm yields a worthwhile improvement.

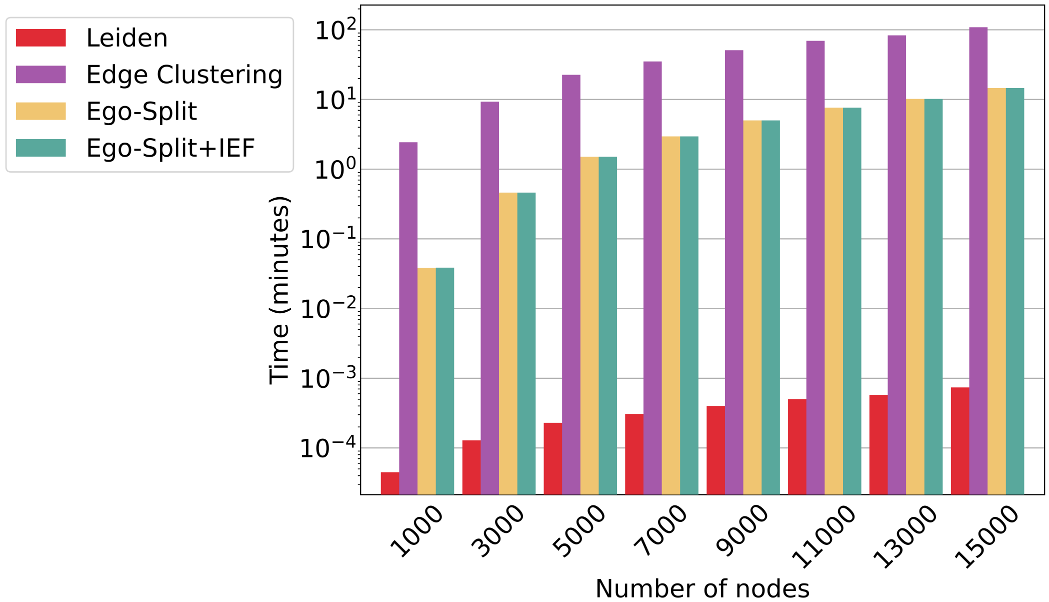

Figure 11: The runtime of four clustering algorithms on **ABCD+o**^{2} graphs with 1,000 to 15,000 nodes.

5 Conclusion

We presented **ABCD+o**^{2}: a generalization of the **ABCD+o** model that allows for overlapping communities. We proposed a novel mechanism for producing overlapping communities, a hidden reference layer, that requires two new parameters: the average number of communities per node η, and the dimension for the reference layer d. Furthermore, we introduced a parameter ρ that controls the correlation between degree and number of communities for non-outlier nodes. In a series of experiments, we compared the properties of the **ABCD+o**^{2} model to those of real graphs using parameters measured from the real graph. We found that the hidden reference layer can accurately create a structure of overlapping communities similar to those in the real graph. We also found that the model can successfully create an empirical ρ close to the requested value (except for extreme cases). Finally, we showcased the model's ability to benchmark community detection algorithms and compare their quality.

This paper acts as a first step in our study of the **ABCD+o**^{2} model and gives way to several open problems for future research. First, a theoretical description of the distribution of the number of communities per node and the effect of the dimension of the reference layer are clear goals. A similar direction is understanding the effects of modifying the geometry of the reference layer. For example, one could estimate the reference space of a network from auxiliary data and use it for a corresponding parameter-fitted **ABCD+o**^{2} model. Using this fitting procedure, can we achieve additional structural similarities to real networks?

We are also interested in results for the **ABCD+o**^{2} model that generalize results of the **ABCD** and **ABCD+o** models. In [20] it was found that the maximum modularity came from the ground truth communities until a certain level of noise, after which a higher modularity could be attained.

One could test for a similar result in $\mathbf{ABCD{+}o}^2$ by comparing the maximum modularity to both the level of noise ξ and the overlap parameter η. A first step to this future research would be selecting an appropriate generalization of modularity for overlapping sets, such as the generalization proposed in [13]. Additionally, in [6] it was shown that **ABCD** graphs exhibit self-similar behaviour, namely, the degree distributions of communities are asymptotically the same as the degree distribution of the whole graph (up to an appropriate normalization). Another open problem is checking if this self-similar property persists in $\mathbf{ABCD{+}o}^2$.

Code Availability

The implementation of the $\mathbf{ABCD{+}o}^2$model is available in julia at `https://github.com/bkamins/ABCDGraphGenerator.jl/`. Code for producing the figures in Sections 3.2 and 4 is available at `https://github.com/ftheberge/ABCDoo_Experiments`.

References

[1] William Aiello, Anthony Bonato, Colin Cooper, Jeanette Janssen, and Paweł Prałat. A spatial web graph model with local influence regions. *Internet Mathematics*, 5(1-2):175–196, 2008.

[2] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. powerlaw: A python package for analysis of heavy-tailed distributions. *PLoS ONE*, 9(1), 2014. `doi:10.1371/journal.pone.0085777`.

[3] Samin Aref, Mahdi Mostajabdaveh, and Hriday Chheda. Bayan algorithm: Detecting communities in networks through exact and approximate optimization of modularity. *Phys. Rev. E*, 110:044315, 2024. `doi:10.1103/PhysRevE.110.044315`.

[4] Albert-László Barabási. *Network science*. Cambridge University Press, 2016. URL: `http://barabasi.com/networksciencebook/`.

[5] Jordan Barrett, Ryan DeWolfe, Bogumił Kamiński, Paweł Prałat, Aaron Smith, and François Théberge. Improving community detection via community association strength scores. In Mindaugas Bloznelis, Paulius Drungilas, Bogumił Kamiński, Paweł Prałat, Matas Šileikis, François Théberge, and Rimantas Vaicekauskas, editors, *Modelling and Mining Networks*, pages 81–95, Cham, 2025. Springer Nature Switzerland. `doi:10.1007/978-3-031-92898-7_6`.

[6] Jordan Barrett, Bogumił Kamiński, Paweł Prałat, and François Théberge. Self-similarity of communities of the ABCD model. *Theoretical Computer Science*, 1026:115012, 2025. `doi:10.1016/j.tcs.2024.115012`.

[7] Edward A Bender and E Rodney Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307, 1978. `doi:10.1016/0097-3165(78)90059-6`.

[8] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975. `doi:10.1145/361002.361007`.

[9] Marian Boguna, Ivan Bonamassa, Manlio De Domenico, Shlomo Havlin, Dmitri Krioukov, and M Ángeles Serrano. Network geometry. *Nature Reviews Physics*, 3(2):114–135, 2021. doi:10.1038/s42254-020-00264-4.

[10] Béla Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980. doi:10.1016/S0195-6698(80)80030-8.

[11] Kyrylo Chykhradze, Anton Korshunov, Nazar Buzun, Roman Pastukhov, Nikolay Kuzyurin, Denis Turdakov, and Hangkyu Kim. Distributed generation of billion-node social graphs with overlapping community structure. In Pierluigi Contucci, Ronaldo Menezes, Andrea Omicini, and Julia Poncela-Casasnovas, editors, *Complex Networks V*, pages 199–208, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-05401-8_19.

[12] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009. doi:10.1137/070710111.

[13] J. Chitra Devi and E. Poovammal. An analysis of overlapping community detection algorithms in social networks. *Procedia Computer Science*, 89:349–358, 2016. Twelfth International Conference on Communication Networks, ICCN 2016, August 19– 21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India. doi:10.1016/j.procs.2016.06.082.

[14] Alessandro Epasto, Silvio Lattanzi, and Renato Paes Leme. Ego-splitting framework: From non-overlapping to overlapping clusters. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 145–154, 2017. doi:10.1145/3097983.3098054.

[15] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010. doi:10.1016/j.physrep.2009.11.002.

[16] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41, 2007. doi:10.1073/pnas.0605965104.

[17] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002. doi:10.1073/pnas.122653799.

[18] Adam Douglas Henry, Paweł Prałat, and Cun-Quan Zhang. Emergence of segregation in evolving social networks. *Proceedings of the National Academy of Sciences*, 108(21):8605–8610, 2011. doi:10.1073/pnas.1014486108.

[19] Bogumił Kamiński, Tomasz Olczak, Bartosz Pankratz, Paweł Prałat, and François Théberge. Properties and performance of the ABCDe random graph model with community structure. *Big Data Research*, 30:100348, 2022. doi:10.1016/j.bdr.2022.100348.

[20] Bogumił Kamiński, Bartosz Pankratz, Paweł Prałat, and François Théberge. Modularity of the ABCD random graph model with community structure. *Journal of Complex Networks*, 10(6):cnac050, 2022. doi:10.1093/comnet/cnac050.

[21] Bogumił Kamiński, Paweł Prałat, and François Théberge. Artificial benchmark for community detection (ABCD)—fast random graph model with community structure. *Network Science*, pages 1–26, 2021. doi:10.1017/nws.2020.45.

[22] Bogumił Kamiński, Paweł Prałat, and François Théberge. *Mining Complex Networks*. Chapman and Hall/CRC, 2021. doi:10.1201/9781003218869.

[23] Bogumił Kamiński, Paweł Prałat, and François Théberge. Artificial benchmark for community detection with outliers (ABCD+o). *Applied Network Science*, 8(1):25, 2023. doi:10.1007/s41109-023-00552-9.

[24] Bogumił Kamiński, Paweł Prałat, and François Théberge. Hypergraph artificial benchmark for community detection (h–ABCD). *Journal of Complex Networks*, 11(4):cnad028, 2023. doi:10.1093/comnet/cnad028.

[25] Paul Kim and Sangwook Kim. Detecting overlapping and hierarchical communities in complex network using interaction-based edge clustering. *Physica A: Statistical Mechanics and its Applications*, 417:46–56, 2015. doi:10.1016/j.physa.2014.09.035.

[26] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009. doi:10.1103/PhysRevE.80.016118.

[27] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008. doi:10.1103/PhysRevE.78.046110.

[28] Aaron F McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*, 2011. doi:10.48550/arXiv.1110.2515.

[29] Günce Keziban Orman and Vincent Labatut. A comparison of community detection algorithms on artificial networks. In *Discovery Science: 12th International Conference, DS 2009, Porto, Portugal, October 3-5, 2009 12*, pages 242–256. Springer, 2009. doi:10.1007/978-3-642-04747-3_20.

[30] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005. doi:10.1038/nature03607.

[31] Neha Sengupta, Michael Hamann, and Dorothea Wagner. Benchmark generator for dynamic overlapping communities in networks. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 415–424, 2017. doi:10.1109/ICDM.2017.51.

[32] M Ángeles Serrano and Marián Boguñá. *The Shortest Path to Network Geometry: A Practical Guide to Basic Models and Applications*. Cambridge University Press, 2021. doi:10.1017/9781108865791.

[33] M Ángeles Serrano, Dmitri Krioukov, and Marián Boguná. Self-similarity of complex networks and hidden metric spaces. *Physical review letters*, 100(7):078701, 2008. doi:10.1103/PhysRevLett.100.078701.

[34] V. Traag, L. Waltman, and Nees Jan van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9:5233, 03 2019. doi:10.1038/s41598-019-41695-z.

[35] N. C. Wormald. *Models of Random Regular Graphs*, page 239–298. London Mathematical Society Lecture Note Series. Cambridge University Press, 1999. doi:10.1017/CBO9780511721335.010.

[36] Nicholas C Wormald. Generating random regular graphs. *Journal of algorithms*, 5(2):247–280, 1984. doi:10.1016/0196-6774(84)90030-0.

[37] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *2012 IEEE 12th International Conference on Data Mining*, pages 745–754, 2012. doi:10.1109/ICDM.2012.138.

[38] Jaewon Yang and Jure Leskovec. Overlapping communities explain core–periphery organization of networks. *Proceedings of the IEEE*, 102(12):1892–1902, 2014. doi:10.1109/JPROC.2014.2364018.

A Density of Intersection Figures

Figures 12, 13 and 14, together with the previously shown Figure 6, present the full experiment comparing community and intersection densities for various ρ. The results are similar to those found in Section 3.2; the density of individual communities in unaffected by ρ, and smaller communities are typically denser than larger communities. However, the density of the intersection does increase with ρ, meaning that dense community intersections may be caused by a positive correlation between degree and the number of communities.

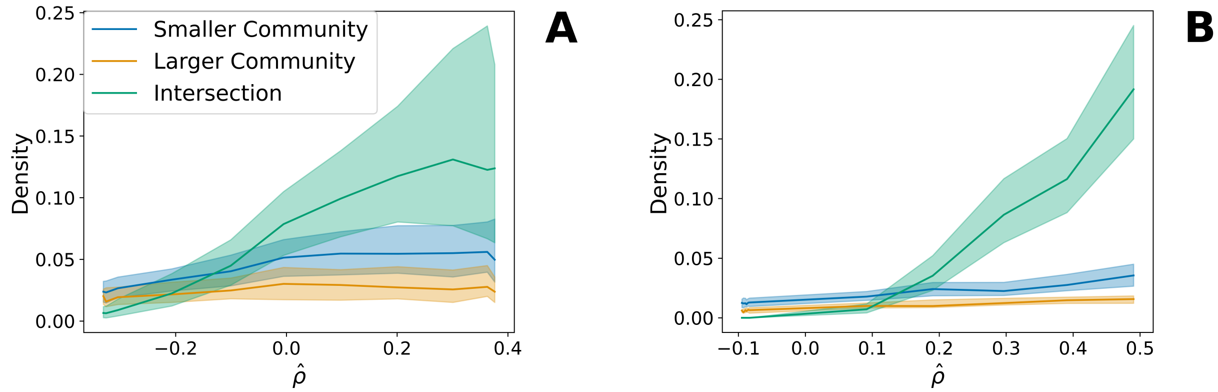

Figure 12: The density of overlaps compared to the empirical ρ on YouTube-like **ABCD+o**^{2} graphs with reference layer dimensions $d = 2$ (panel A) and $d = 64$ (panel B).

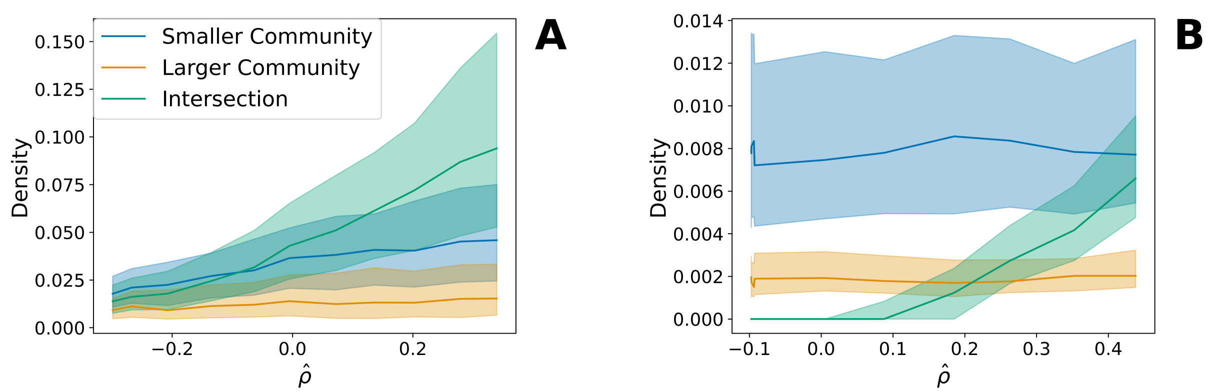

Figure 13: The density of overlaps compared to the empirical ρ on DBLP-like **ABCD+o**^{2} graphs with reference layer dimensions $d = 2$ (panel A) and $d = 64$ (panel B).

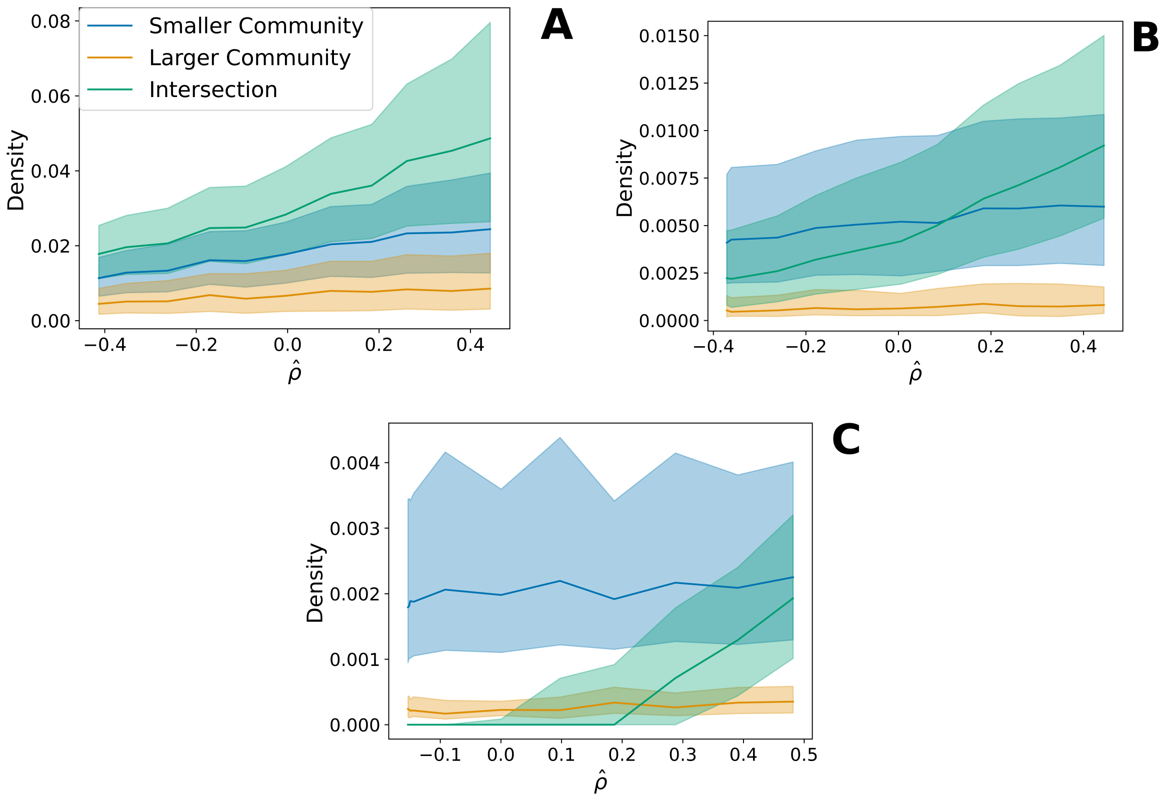

Figure 14: The density of overlaps compared to the empirical ρ on Amazon-like **ABCD+o**^{2} graphs with reference layer dimensions $d = 2$ (panel A), $d = 8$ (panel B) and $d = 64$ (panel C).

B Internal Edge Fraction Figures

Figures 15, 16 and 17, together with the previously shown Figure 8, present the full experiment showing the range of the top five internal edge fractions, sorted in descending order, on the real networks (left) and the corresponding **ABCD+o**^{2} graphs (right), binned by the number of communities the node v is in. The results are similar to those found in Section 3.2, with a node in ℓ communities typically having ℓ large but unequal IEF values. Furthermore, the gap between the ℓ and $\ell + 1$ IEF values tends to decrease as ℓ increases.

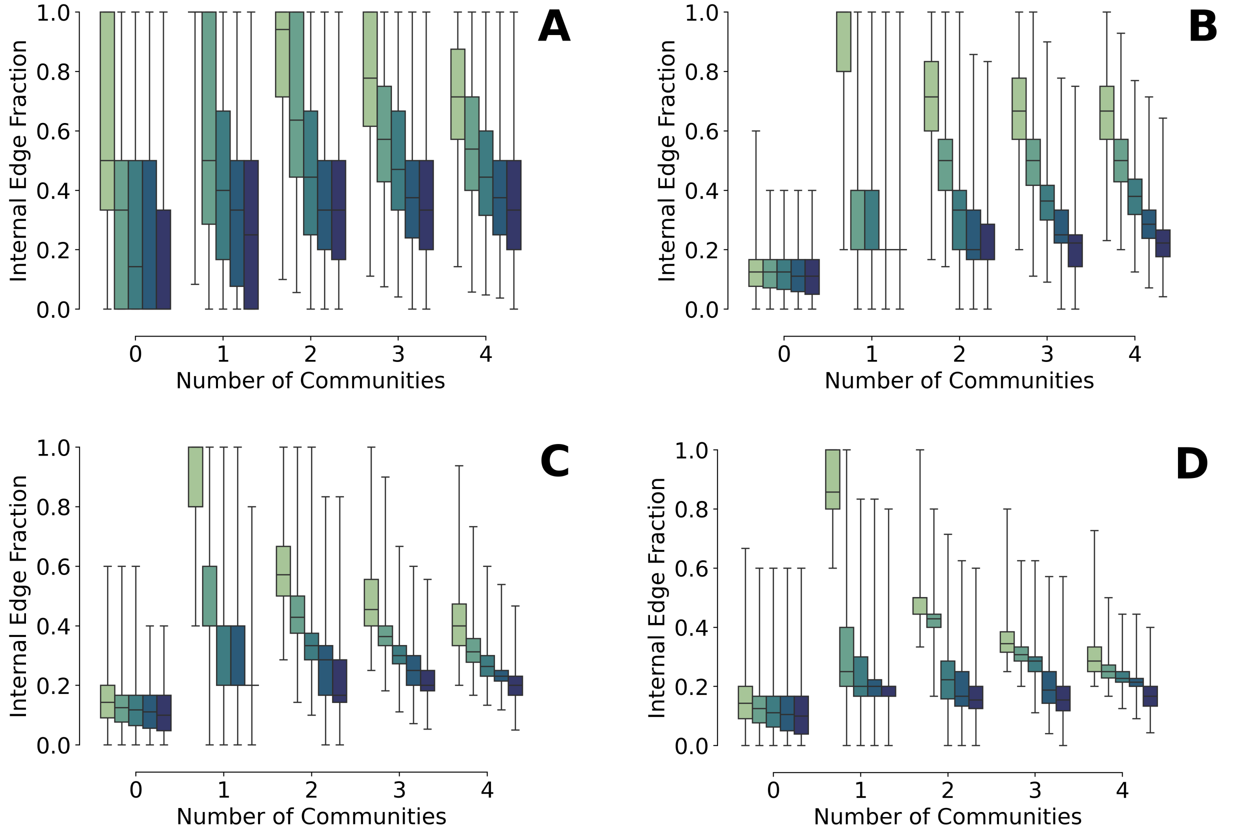

Figure 15: The top 5 IEF values in decreasing order, grouped by the number of communities the node belongs to, for DBLP and DBLP-like **ABCD+o**^{2}graphs. Panel A corresponds to the real graph, and panels B, C, D correspond to the synthetic counterparts with reference layer dimensions $d = 2$, $d = 8$, and $d = 64$ respectively. Panels A and C correspond to panels A and B of Figure 8, but are repeated here for easy comparison with the other reference layer dimensions.

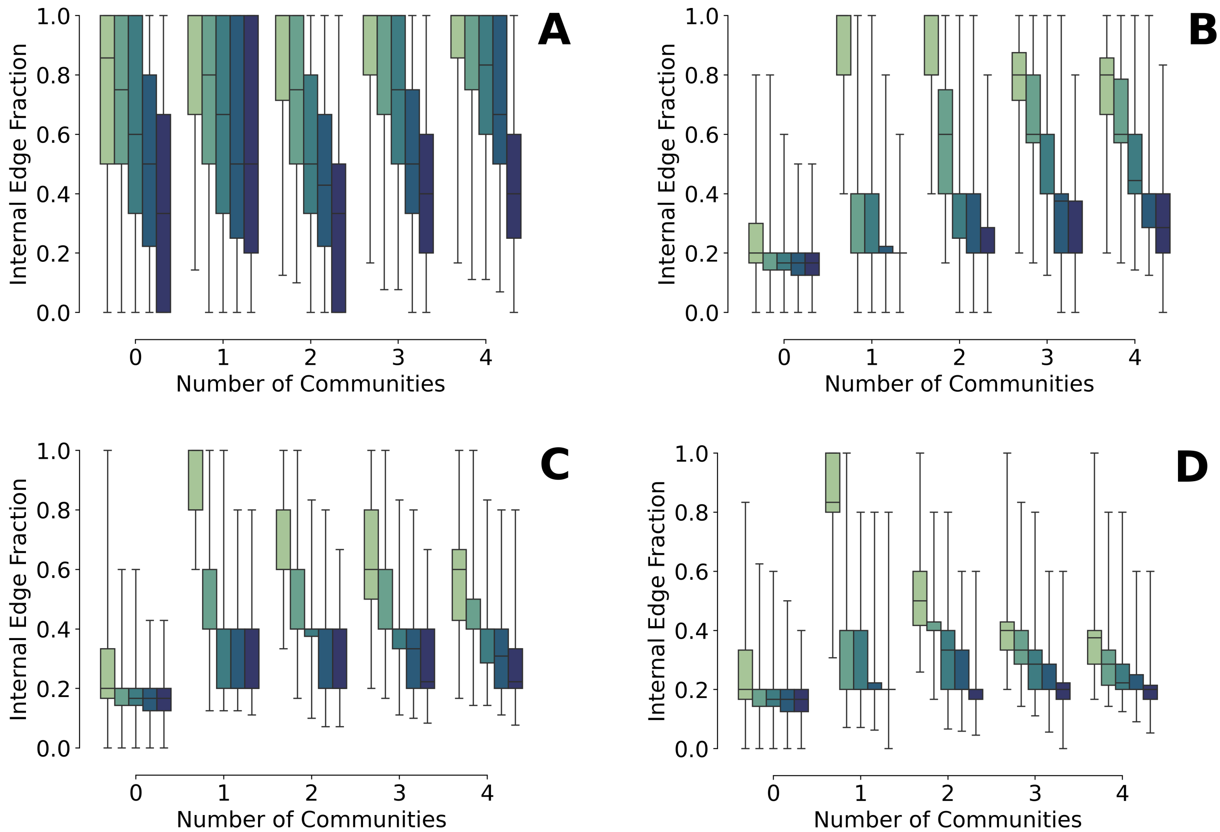

Figure 16: The top 5 IEF values in decreasing order, grouped by the number of communities the node belongs to, for Amazon and Amazon-like **ABCD+o**^{2}graphs. Panel A corresponds to the real graph, and panels B, C, D correspond to the synthetic counterparts with reference layer dimensions $d = 2$, $d = 8$, and $d = 64$ respectively

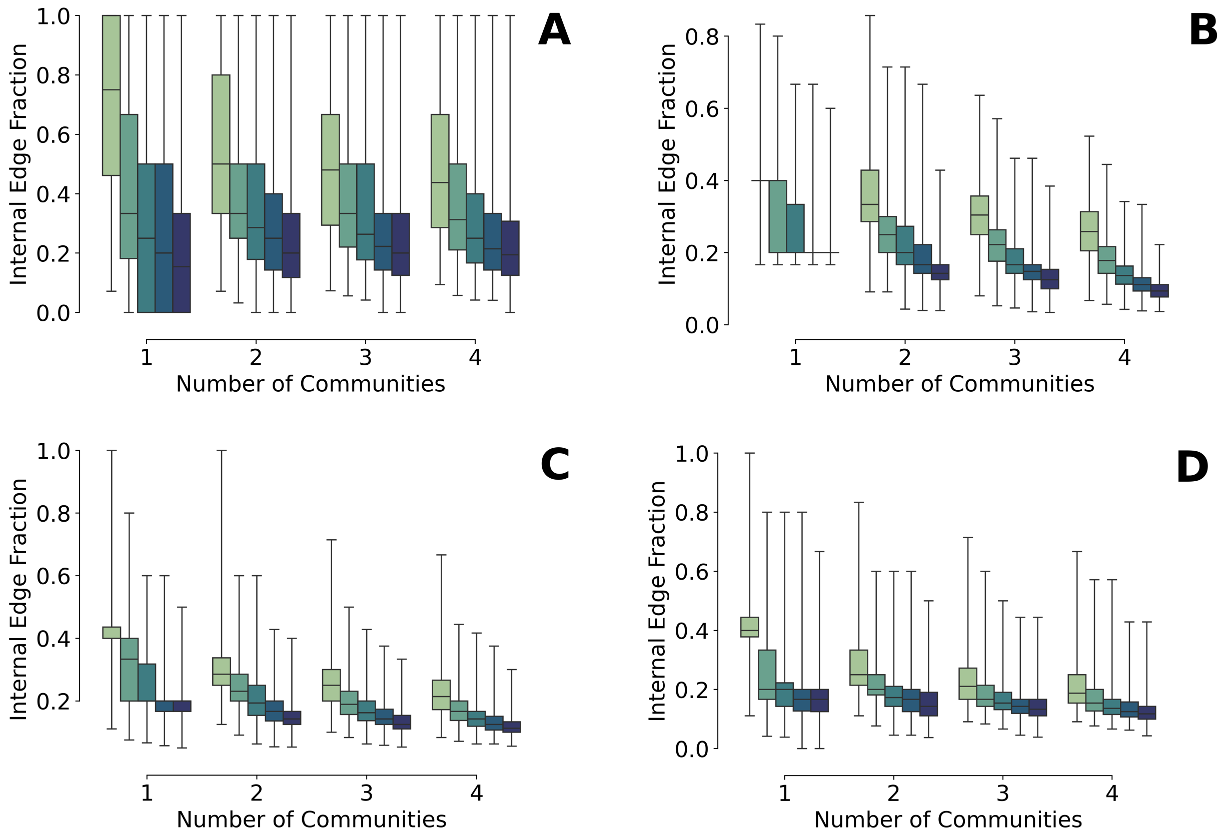

Figure 17: The top 5 IEF values in decreasing order, grouped by the number of communities the node belongs to, for YouTube and YouTube-like **ABCD+o**^{2}graphs. Panel A corresponds to the real graph, and panels B, C, D correspond to the synthetic counterparts with reference layer dimensions $d = 2$, $d = 8$, and $d = 64$ respectively