Sallmen, Sallamari; Nurmi, Tarmo; Kivelä, Mikko

Graphlets in multilayer networks

Published in:
Journal of Complex Networks

DOI:
10.1093/comnet/cnac005

Published: 06/04/2022

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Sallmen, S., Nurmi, T., & Kivelä, M. (2022). Graphlets in multilayer networks. *Journal of Complex Networks*, *10*(2), Article cnac005. https://doi.org/10.1093/comnet/cnac005

Graphlets in multilayer networks

Sallamari Sallmen* Tarmo Nurmi*† Mikko Kivelä*

June 7, 2022

Abstract

Representing various networked data as multiplex networks, networks of networks and other multilayer networks can reveal completely new types of structures in these system. We introduce a general and principled graphlet framework for multilayer networks which allows one to break any multilayer network into small multilayered building blocks. These multilayer graphlets can be either analyzed themselves or used to do tasks such as comparing different systems. The method is flexible in terms of multilayer isomorphism, automorphism orbit definition, and the type of multilayer network. We illustrate our method for multiplex networks and show how it can be used to distinguish networks produced with multiple models from each other in an unsupervised way. In addition, we include an automatic way of generating the hundreds of dependency equations between the orbit counts needed to remove redundant orbit counts. The framework introduced here allows one to analyze multilayer networks with versatile semantics, and these methods can thus be used to analyze the structural building blocks of myriad multilayer networks.

1 Introduction

Representing networked systems as graphs has been an extremely successful approach for analyzing the structure of various such systems, ranging from societies and transportation systems to brains and cellular regulation [1]. One of the reasons for the rapid growth of structural network analysis is that, building on the graph abstraction, it has been possible to analyze the structure of these otherwise disparate systems with the same methods and models. Despite this success, there are several systems and research questions for which one needs to consider more general network structures, such as multilayer networks [2, 3]. Several methods and concepts have been generalized in order to understand a wide class of multilayered networks, including community detection methods [4, 5, 6], spreading processes [7], centrality measures [8, 9], and local network features [10, 11]. As multilayer networks are higher order structures than graphs, these generalizations typically have more degrees of freedom in their parametrization, and many concepts can be generalized in multiple different ways. In order to understand which of these generalizations to use, or how to navigate the

*Department of Computer Science, Aalto University School of Science, P.O. Box 15400, FI-00076, Finland
†tarmo.nurmi@aalto.fi

new degrees of freedom, it is useful to take a principled approach where complicated concepts are built bottom-up from more fundamental ones.

Recently, the concept of graph isomorphism, a notion formalizing structural similarity, was extended to multilayer networks [12]. Graph isomorphism is a fundamental idea behind structural analytics such as graphlets, motifs, and network comparison. When generalized to multilayer networks, these concepts can then be used to analyze various types of multilayer networks coming from multiple application areas. Further, additional layers of methodology can be developed. Here we focus on one such methodology known as *graphlets*, which have several applications in network data analysis. They reduce the topological information on single nodes or complete networks into vectorized format, which can be used, for example, in supervised and unsupervised learning tasks. A common use case is to compute topological similarity and dissimilarity of pairs of networks using graphlets. This type of alignment-free methods [13] are less computationally intensive than methods where the full network would need to be aligned [14, 15]. Further, they can be used to group together networks that share common features, for example, because they are created via similar mechanisms [16, 17, 18, 14]. Graphlet-based methods are built on *graphlet degree distributions*, which are generalizations of the degree distribution of a network [16]. For example, they are used to assess the distance between two different networks [17, 14, 13]. Graphlet degrees for nodes can be defined based on the full network [16, 17, 14] or on the ego-networks of nodes [18]. Mathematically, the graphlet degree of a node is defined by how many times it is found on a specific *automorphism orbit*, a node equivalence class based on network automorphisms within the graphlets of the network (*orbit* for short).

While graphlets and orbits have been defined and studied in graphs, analysis for multilayer networks is not fully developed. Graphlets in multiplex networks, which is an important special case of multilayer networks [2], have been studied recently [19]. Orbits of nodes in multiplex graphlets were defined as "sub-orbits" of aggregated single-layer graphlet orbits, where the specific multiplex edge contents of the aggregated graphlets classify the nodes into the sub-orbits. Additionally, these multiplex graphlets can be grouped together using various reduction strategies in order to decrease the number of sub-orbits [19]. However, the sub-orbit and reduction approach is not based on the explicit definition of orbits via multilayer network automorphism groups. This means that it will catch only a subset of all isomorphisms available for multiplex networks. Multiplex graphlets have also been defined for three nodes in two-layer networks [20] and multiplex motifs (statistically enriched graphlets) have been defined for larger numbers of nodes and layers as manually constructed compositions of smaller subgraphs [21], using node isomorphism as the isomorphism type of choice explicitly or implicitly (node isomorphism is only one of the many types of isomorphisms possible in multiplex networks [12]).

In this paper, we develop multilayer definition of graphlets and related concepts needed for comparing multilayer networks via graphlet degree distributions. Our definition applies to general multilayer networks with any number of aspects [2]. It is based on an explicit definition of multilayer network isomorphism [12], and we use a multilayer automorphism group as a basis of our definition of the multilayer orbits. We apply our framework to a special case of multiplex networks in order to illustrate the concept of multilayer graphlets and to compare it to the recently developed multiplex graphlets [19], which results in a different orbit definition (see Supplementary Materials section A). In addition, we define a procedure to automatically produce multiplex dependency equations [17] for the graphlet

degrees for any choice of isomorphism, number of nodes, and number of layers. In contrast to previous methods for analysing small substructures of multilayer networks [22, 23, 24, 25, 26], our definition of multilayer graphlets and orbits captures graphlets with any number of nodes, layers and aspects in addition to all multilayer isomorphism types. That is, instead of suggesting a single-purpose method, our method includes considerable and transparent freedom to choose how the graphlet analysis should be extended to various use cases.

The additional choices and multilayer extensions one needs to make when working with multilayer graphlets are illustrated clearly in the task of comparing multilayer networks (see Figure 1). This article is organised such that we build up this pipeline starting from the basics, and divide the text into the following aims: 1) Defining automorphism orbits in multilayer networks by extending the notion of conventional automorphism orbits; 2) Defining multilayer graphlets in order to apply the orbit definition to them; 3) Using the previous two to construct multilayer versions of graphlet/orbit-based alignment-free network distance measures; 4) Illustrating the multilayer pipeline on a simplified type of multiplex networks, and explicitly constructing dependency equations that quantify how orbit counts depend on other orbit counts in them (the removal of redundant orbit counts has been used before for single-layer networks, where it was found to improve the performance of a graphlet-based distance measure in some cases [17]); Finally, 5) Evaluating different distance measures on test sets of multiplex network models. The pipeline for creating multiplex orbit dependency equations, calculating multiplex graphlet degrees and creating Figures 5–7 is implemented by the authors and is publicly available [27].

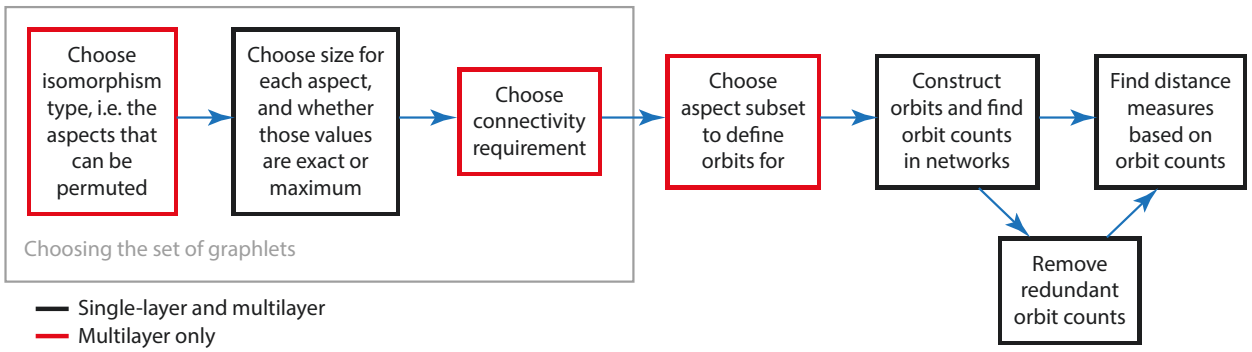


Figure 1: The pipelines for single-layer and multilayer graphlet/orbit-based network distance measure calculation. The black boxes depict steps that have to be taken in both the single-layer and multilayer pipeline, while the red boxes depict steps unique to the multilayer pipeline. A typical pipeline for calculating graphlet/orbit-based distance measures in single-layer networks starts with deciding how many nodes the graphlets should have. The graphlets are then constructed, orbits are found and enumerated from the networks in question, and the distance measures are calculated based on those orbit counts. The orbit counts contain redundancies because some graphlets contain other graphlets as sub-graphs; therefore, if desired, the level of redundancy can be reduced by removing some of the redundant orbit counts prior to distance measure calculation. In multilayer networks, the pipeline involves more choices. Now, we have to decide the type of multilayer isomorphism that is used to define the graphlets and orbits, the graphlet size for each aspect (and whether that size means exactly that many or at most that many), the definition of when a graphlet is "connected" in the multilayer setting, and the subset of aspects we want to find the orbits for.

2 Framework for multilayer network graphlet analysis

We start by defining the multilayer graphlet framework. As the multilayer graphlets are an extension of graphlets in ordinary single-layer graphs, we first review how automorphism orbits are defined in graphs. Then, we extend this definition to multilayer networks by incorporating multilayer network isomorphisms in place of graph isomorphisms. Then, we define graphlets in multilayer networks and the automorphism orbits within them. Finally, we generalize some existing graphlet-based distance measures for multilayer networks.

2.1 Automorphism orbits

2.1.1 Ordinary graphs

Graph isomorphism is a concept defining when two graphs are structurally equivalent. Informally, one can think of two graphs being isomorphic if they can be drawn in exactly the same way (while disregarding the node names). Formally, two graphs are isomorphic if one can transform one of the graphs into the other by relabeling the nodes. A permutation of node labels that performs the transformation is called an isomorphism, and an isomorphism from a graph to itself is called an automorphism. In other words, if a relabeling g assigns (maps) new identities to the nodes such that the set of edges in the network is the exact same before and after applying it, g is an automorphism. The relabeling does not need to change the identity of each node: in fact, the relabeling that maps each node to itself is called the identity permutation.

Note that if you relabel nodes of a graph G with a permutation g that is an automorphism and then with another automorphism f, then this combination fg is another permutation that is also an automorphism for G. That is, the automorphisms of a graph are related to each other. The structure of these relationships can be studied algebraically by forming a group of all automorphisms and the combination relation. This group is known as the *automorphism group* of the graph denoted by $Aut(G)$ [16]. These automorphisms partition the nodes of the graph into equivalence classes called automorphism orbits. Formally, for a graph $G = (V, E)$, where V and E are the sets of nodes and edges, the automorphism orbit of node u is defined as the set of all the nodes it can be mapped to with any automorphism [16]

$$Orb(u) \;=\; \{v \mid v = g(u)\ \exists\ g \in Aut(G)\}\,. \tag{1}$$

One can interpret the automorphism orbits (or orbits for short) as sets of nodes that are structurally equivalent in the given graph.

2.1.2 Multilayer networks

A multilayer network is a quadruplet $M = (V_M, E_M, V, \mathbf{L})$ where $\mathbf{L} = \{L_a\}_{a=1}^{d}$ is a sequence of sets of elementary layers, V_M is the set of node-layers, E_M is the set of edges between them ($E_M \subseteq V_M \times V_M$), and V is the set of nodes [2]. Elementary layers are the basic elements of the layers in the network, and a layer is a collection of elementary layers, one elementary layer for each $a = 1, 2, ..., d$. Thus, in order to define a layer, d elementary layers have to be specified. Nodes reside on layers, and the combination of a node along with the layer it

appears on is called a node-layer. The number of aspects in the network, d, corresponds to the "dimensionality" of the layers $\boldsymbol{\alpha} \in L_1 \times L_2 \times ... \times L_d$ in the network. Each node-layer is a combination of a node identity $v \in V$ and layer identity $\boldsymbol{\alpha}$, and $V_M \subseteq V \times L_1 \times L_2 \times ... \times L_d$. For notational convenience we will denote $L_0 = V$, i.e. define the elementary layers of the 0th aspect as the node identities [12]. We will denote node-layers as $(v, \boldsymbol{\alpha}) \in V_M$ or $(\gamma_0, \ldots, \gamma_d) \in V_M$. The graph formed by the combination (V_M, E_M) is called the underlying graph of the multilayer network.

The top left of Figure 2 illustrates a multilayer network. In that network, there are two nodes (1 and 2), two elementary layers in the first aspect (x and y), and two elementary layers in the second aspect (α and β). The network is thus a two-aspect network. The blue rectangles correspond to layers, and to define a layer, an elementary layer from both the first and the second aspect is required (consequently, the dimensionality of the layers is equal to the number of aspects). There are four layers in total: (x, α), (y, α), (x, β), and (y, β). Nodes 1 and 2 are found on each of the four layers (in general, not all nodes have to necessarily reside on every layer in multilayer networks). When a node exists on a layer, the combination of a node and the layer it exists on is called a node-layer. There are eight node-layers in the network, $(1, x, \alpha)$, $(2, x, \alpha)$, $(1, y, \alpha)$, $(2, y, \alpha)$, $(1, x, \beta)$, $(2, x, \beta)$, $(1, y, \beta)$, and $(2, y, \beta)$. Edges are defined between these node-layers and can be classified into edges that are between nodes that exists on the same layer (intralayer edges, of which there are none in this network) and between nodes that exist on different layers (interlayer edges). To make the example more concrete, such a two-aspect network could be thought to represent publication activity of two scientist who work in multiple fields: 1 and 2 are two scientists, x and y are two scientific disciplines, and α and β are two different years. The node-layer $(1, x, \alpha)$ could then represent that scientist 1 has actively published in discipline x during year α. An edge between two node-layers would signify that there is a cross-reference between those publication activities. A person can cross-reference themselves across disciplines and years, or two people can cross-reference each other during the same year but in different disciplines, etc., all of which are allowed by the multilayer network definition.

In multilayer networks the isomorphisms and automorphisms depend on the aspects p that are allowed to be permuted [12]. For a network with d aspects, $p \subseteq \{0, 1, 2, ..., d\}$. The isomorphism of choice (the set of aspects to be permuted) affects the orbits in multilayer networks. The automorphisms of multilayer network M with p as the set of aspects that are allowed to be permuted form the p-automorphism group $Aut_p(M)$. Let $\boldsymbol{\zeta} \in Aut_p(M)$ be a p-automorphism (relabeling of nodes and elementary layers) of M. The total relabeling consists of a relabeling for each aspect: $\boldsymbol{\zeta} = (\zeta_0, \zeta_1, ..., \zeta_d)$, where $\zeta_a = \mathbb{1}_{L_a}$ if $a \notin p$ and $\mathbb{1}_{L_a}$ is the identity permutation for the set of elementary layers L_a of aspect a. For node u in a multilayer network M the p-automorphism orbit is defined analogous of Equation 1 as

$$Orb_p(u) = \{v \mid v = \zeta_0(u) \; \exists \; \boldsymbol{\zeta} \in Aut_p(M)\}, \text{ where } \boldsymbol{\zeta} = (\zeta_0, \zeta_1, ..., \zeta_d)\,. \tag{2}$$

In addition to defining the orbits for nodes (as in ordinary graphs), in multilayer networks it is also possible to define the orbits for node-layers, layers, or any other subset of aspects. For node-layer $(u, \boldsymbol{\beta})$ the orbit is defined as

$$Orb_p((u, \boldsymbol{\beta})) = \{(v, \boldsymbol{\alpha}) \mid (v, \boldsymbol{\alpha}) = \boldsymbol{\zeta}(u, \boldsymbol{\beta}) \; \exists \; \boldsymbol{\zeta} \in Aut_p(M)\}\,. \tag{3}$$

We can define the orbits for any subset of aspects: let $\boldsymbol{\gamma} \in L_{a_1} \times L_{a_2} \times ... \times L_{a_k}$, where

$a_1, ..., a_k$ are the desired aspects. Then,

$$Orb_p(\boldsymbol{\gamma}) \;=\; \{\boldsymbol{\delta} \mid \delta_1 = \zeta_{a_1}(\gamma_1), \delta_2 = \zeta_{a_2}(\gamma_2), ..., \delta_k = \zeta_{a_k}(\gamma_k) \;\exists\; \boldsymbol{\zeta} \in Aut_p(M)\}\ . \qquad (4)$$

Equation 4 becomes Equation 2 when we choose $k = 1, a_1 = 0$ ($L_0 = V$), and Equation 3 when we choose $k = d+1, a_1 = 0, a_2 = 1, a_3 = 2, ..., a_k = d$. As in ordinary graphs, the orbits in multilayer networks partition the nodes, node-layers, or the entities $\boldsymbol{\gamma}$ of any subset of aspects into disjoint equivalence classes. This follows from the fact that the application of an automorphism is a group action and we can apply the known result that group action induces equivalence relation [28] with the equivalence class of $\boldsymbol{\gamma}$ that is $Orb_p(\boldsymbol{\gamma})$. For an explicit proof involving the properties of multilayer automorphisms, see Supplementary Materials section B.

Figure 2 illustrates all the automorphisms, orbits and orbit equivalence classes in a small two-aspect multilayer network.

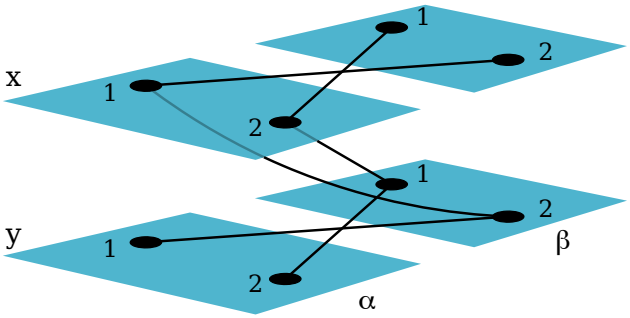
2.2 Graphlets and graphlet degrees

Graphlets in ordinary single-layer networks are defined as small, connected, non-isomorphic induced subgraphs of a larger network [29, 16]. A graphlet is thus an isomorphism class of connected induced subgraphs. Graphlet analysis is usually restricted to a subset of all possible graphlets, for example by looking only at graphlets with at most some number of nodes. While the graphlet definition is quite straight-forward in single-layer networks, it leads to different definitions of graphlets in multilayer networks based on how one defines isomorphism, connectivity, and size. Figure 1 illustrates the additional choices to be made in graphlet definition in the multilayer setting compared to the single-layer case. We now present a formal definition of multilayer graphlets, elaborate on the concepts of isomorphism, connectivity, and size, and define graphlet degrees in multilayer networks.

Formally, an induced subnetwork $M' = (V'_M, E'_M, V', \mathbf{L}')$ within a larger network $M = (V_M, E_M, V, \mathbf{L})$ is defined by $V' \subseteq V$, $L'_a \subseteq L_a \;\forall\; a \in \{1, 2, ..., d\}$, $V'_M = \{(v, \boldsymbol{\alpha}) \in V_M \mid (v, \boldsymbol{\alpha}) \in V' \times L'_1 \times L'_2 \times ... \times L'_d\}$, $E'_M = \{((v, \boldsymbol{\alpha}), (u, \boldsymbol{\beta})) \in E_M \mid ((v, \boldsymbol{\alpha}), (u, \boldsymbol{\beta})) \in V'_M \times V'_M\}$. This definition fixes the elementary layer sets for each aspect, and then all the node-layers and edges that exist within the span of those elementary layer sets are included. M' belongs in an isomorphism class with every other subnetwork it is isomorphic to. If the subnetworks in that isomorphism class are connected, we call that isomorphism class the graphlet that M' corresponds to.

Unlike single-layer networks, multilayer networks have multiple possible types of isomorphism, one for each set $p \subseteq \{0, 1, 2, ..., d\}$ of aspects that can be permuted. The choice of isomorphism affects the set of graphlets in a given network and influences which subnetworks correspond to the same graphlet. Which isomorphism is appropriate depends on the application [12].

When it comes to connectivity, one option is to require the underlying graph of the graphlet to be connected. However, this could lead to some entities in the network never participating in any graphlets, and one may wish to loosen the restriction to requiring only the layer-aggregated network to be connected. For example, if node-layer $(v, \boldsymbol{\alpha})$ is not connected to any other node-layer, then node v does not participate in any graphlets in subnetworks that include layer $\boldsymbol{\alpha}$ if the underlying graph is required to be connected. However,



$$
\begin{aligned}
Aut_{\{0\}} &= \{\mathbb{1}, (1 \leftrightarrow 2, \mathbb{1}_{L_1}, \mathbb{1}_{L_2})\} \\
Aut_{\{1\}} &= \{\mathbb{1}\} \\
Aut_{\{2\}} &= \{\mathbb{1}\} \\
Aut_{\{0,1\}} &= \{\mathbb{1}, (1 \leftrightarrow 2, \mathbb{1}_{L_1}, \mathbb{1}_{L_2})\} \\
Aut_{\{0,2\}} &= \{\mathbb{1}, (1 \leftrightarrow 2, \mathbb{1}_{L_1}, \mathbb{1}_{L_2})\} \\
Aut_{\{1,2\}} &= \{\mathbb{1}, (\mathbb{1}_{L_0}, x \leftrightarrow y, \alpha \leftrightarrow \beta)\} \\
Aut_{\{0,1,2\}} &= \{\mathbb{1}, (1 \leftrightarrow 2, \mathbb{1}_{L_1}, \mathbb{1}_{L_2}), \\
&\qquad (\mathbb{1}_{L_0}, x \leftrightarrow y, \alpha \leftrightarrow \beta), \\
&\qquad (1 \leftrightarrow 2, x \leftrightarrow y, \alpha \leftrightarrow \beta)\}
\end{aligned}
$$

$Orb_p(\boldsymbol{\gamma})$ equivalence classes ($\boldsymbol{\gamma} \in L_{a_1} \times \ldots \times L_{a_k}$)

p \ $a_1, \ldots, a_k$	0	1	2	0, 1	0, 2	1, 2	0, 1, 2
$\{0\}$, $\{0,1\}$, and $\{0,2\}$	$\{1,2\}$	$\{x\}$, $\{y\}$	$\{\alpha\}$, $\{\beta\}$	$\{(1,x),(2,x)\}$, $\{(1,y),(2,y)\}$	$\{(1,\alpha),(2,\alpha)\}$, $\{(1,\beta),(2,\beta)\}$	$\{(x,\alpha)\}$, $\{(y,\alpha)\}$, $\{(x,\beta)\}$, $\{(y,\beta)\}$	$\{(1,x,\alpha),(2,x,\alpha)\}$, $\{(1,y,\alpha),(2,y,\alpha)\}$, $\{(1,x,\beta),(2,x,\beta)\}$, $\{(1,y,\beta),(2,y,\beta)\}$
$\{1,2\}$	$\{1\}$, $\{2\}$	$\{x,y\}$	$\{\alpha,\beta\}$	$\{(1,x),(1,y)\}$, $\{(2,x),(2,y)\}$	$\{(1,\alpha),(1,\beta)\}$, $\{(2,\alpha),(2,\beta)\}$	$\{(x,\alpha),(y,\beta)\}$, $\{(x,\beta),(y,\alpha)\}$	$\{(1,x,\alpha),(1,y,\beta)\}$, $\{(1,y,\alpha),(1,x,\beta)\}$, $\{(2,x,\alpha),(2,y,\beta)\}$, $\{(2,y,\alpha),(2,x,\beta)\}$
$\{0,1,2\}$	$\{1,2\}$	$\{x,y\}$	$\{\alpha,\beta\}$	$\{(1,x),(2,x),$ $(1,y),(2,y)\}$	$\{(1,\alpha),(2,\alpha),$ $(1,\beta),(2,\beta)\}$	$\{(x,\alpha),(y,\beta)\}$, $\{(x,\beta),(y,\alpha)\}$	$\{(1,x,\alpha),(2,x,\alpha),$ $(1,y,\beta),(2,y,\beta)\}$, $\{(1,y,\alpha),(2,y,\alpha),$ $(1,x,\beta),(2,x,\beta)\}$

Figure 2: **Top left:** A two-aspect multilayer network with $L_0 = V = \{1,2\}$, $L_1 = \{x,y\}$, $L_2 = \{\alpha,\beta\}$. **Top right:** The automorphism groups of the network on the left, where $\mathbb{1} = (\mathbb{1}_{L_0}, \mathbb{1}_{L_1}, \mathbb{1}_{L_2})$ is the identity permutation in every aspect. In the non-identity permutations, $\square \leftrightarrow \lozenge$ denotes that $\square$ is relabeled to $\lozenge$ and $\lozenge$ is relabeled to $\square$. Note that in general the p-automorphism group cannot be inferred from the automorphism groups of subsets of p [12]: $Aut_{\{1\}}$ and $Aut_{\{2\}}$ contain only the identity permutation, but $Aut_{\{1,2\}}$ contains also another permutation. **Bottom:** The orbit equivalence classes for the network on the top left. When $p = \{1\}$ or $p = \{2\}$, each entity $\boldsymbol{\gamma}$ is alone in its own equivalence class regardless of $a_1, \ldots, a_k$; therefore, these have been omitted from the table. If there is more than one equivalence class, different colors have been used to visually separate them. An entity is always in its own equivalence class, so the table can be used to find the orbit of each entity: for example, $Orb_{\{0\}}(1) = \{1,2\} = Orb_{\{0\}}(2)$ and $Orb_{\{0,1,2\}}((x,\alpha)) = \{(x,\alpha),(y,\beta)\} = Orb_{\{0,1,2\}}((y,\beta))$.

if only the aggregated graph is required to be connected, v can still participate in graphlets if there are sufficient connections on other layers.

The size of a multilayer graphlet can be defined based on, for example, the number of nodes, layers, or node-layers participating in the graphlet. A reasonable extension of the notion of graphlet size is that we give the size of the graphlet in every aspect (including the zeroth aspect of nodes), which means that in total we need $d+1$ numbers to define the graphlet size. In single-layer networks this would be just one number, the number of nodes, in one-aspect multilayer networks this would be the number of nodes and the number of layers, in two-aspect networks this would be the number of nodes, the number of elementary layers in the first aspect, and the number of elementary layers in the second aspect, and so on. According to this definition, the size of the graphlet that an induced subnetwork $M' = (V'_M, E'_M, V', \mathbf{L}')$ corresponds to is then $|V'|$, $|L'_1|$, $|L'_2|$, ..., $|L'_d|$. Notably, this definition of size does not fix the size of V'_M: two graphlets with the same size can contain a different number of node-layers.

Graphlet degrees

Graphlets are intertwined with the concept of node roles and automorphism orbits in single-layer networks, such that the automorphism orbit of a node within a graphlet can be used to define the node's role in that graphlet [17]. This concept can be immediately generalized to multilayer networks by applying the multilayer automorphism node orbit definition (Equation 2) in place of the single-layer automorphism orbit. The number of times a node is found on a specific orbit of a specific graphlet in a network is called the *graphlet degree* of that node with respect to that orbit [16]. The distribution of graphlet degrees of a specific orbit over all the nodes in a network is called the *graphlet degree distribution* of that orbit [16] — there is one such distribution for each orbit, both in an ordinary graph and in a multilayer network (naturally, the orbits themselves will be different in the two cases). In addition to defining graphlet degrees of nodes, in multilayer networks we can define a graphlet degree with respect to layers, node-layers, or any other subset of aspects, since orbits are defined for all of them (see the previous section). The graphlet degree of $\boldsymbol{\gamma} \in L_{a_1} \times L_{a_2} \times ... \times L_{a_k}$ w.r.t. an orbit of a graphlet in a multilayer network M, where $a_1, ..., a_k$ are aspects of M, is then simply the number of times $\boldsymbol{\gamma}$ is found on that specific orbit in M. Figure 3 illustrates graphlet degrees for a combination of a node and an elementary layer in a two-aspect multilayer network with respect to graphlets of certain size. Because of the added degrees of freedom in multilayer networks compared to ordinary graphs, when talking about graphlet degrees one needs to specify which combination of aspects and which isomorphism is considered.

When finding the graphlets contained in a network, we need to choose the type of isomorphism, and when determining the orbits inside those graphlets, we need to choose the type of automorphism. Both of these require the choice of which aspects are allowed to be permuted, and it is reasonable to use the same set of aspects for both. However, this is not required and they can be different, in which case one needs to be careful of the interpretations of the real-world meaning of the graphlet degree distributions.

As is the case with single-layer graphlets, multilayer graphlets contain smaller graphlets as subnetworks, and therefore there are dependencies between the graphlet degrees of different orbits. Similar to single-layer networks [17, 30], it is possible to construct orbit count equations that exactly determine these dependencies in multilayer networks. In the mul-

tiplex network case study in this paper, we describe a process of automatically generating dependency equations for single-aspect multiplex networks in detail.

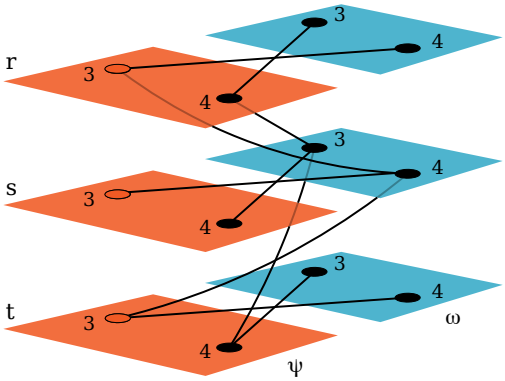


Figure 3: A two-aspect multilayer network with $L_0 = V = \{3, 4\}$, $L_1 = \{r, s, t\}$, $L_2 = \{\psi, \omega\}$. We are interested in the graphlet degrees of $(3, \psi)$ (the corresponding node and the layers are shown in orange) with respect to graphlets of size (2, 2, 2). There are three induced subnetworks of that size, $(\{3, 4\}, \{r, s\}, \{\psi, \omega\})$, $(\{3, 4\}, \{s, t\}, \{\psi, \omega\})$, and $(\{3, 4\}, \{r, t\}, \{\psi, \omega\})$. Depending on the connectivity requirements, $(\{3, 4\}, \{r, t\}, \{\psi, \omega\})$ may or may not be connected and thus it may or may not be a graphlet according to the definition. If the set of aspects that can be permuted, p, is $\{1\}$, $\{0, 1\}$, $\{1, 2\}$ or $\{0, 1, 2\}$, then $(\{3, 4\}, \{r, s\}, \{\psi, \omega\})$ and $(\{3, 4\}, \{s, t\}, \{\psi, \omega\})$ are isomorphic and thus correspond to the same graphlet, and $(3, \psi)$ is in the same orbit in both of them. The graphlet degree of $(3, \psi)$ with respect to that orbit of that graphlet is then 2. If $(\{3, 4\}, \{r, t\}, \{\psi, \omega\})$ is considered connected, it corresponds to a different graphlet (it is not isomorphic to either of the two other subnetworks) and thus the graphlet degree of $(3, \psi)$ with respect to the orbit it is on in that graphlet is 1. If $p \in \{\{0\}, \{2\}, \{0, 2\}\}$, then none of the induced subnetworks are isomorphic to the other(s) and every one of them corresponds to a different graphlet. For each of these graphlets, $(3, \psi)$ is the only element in its orbit in that graphlet, and thus the graphlet degree of $(3, \psi)$ with respect to each of those orbits is 1. If $(\{3, 4\}, \{r, t\}, \{\psi, \omega\})$ is considered connected, there are three different graphlet degrees equal to 1, and if it is not, there are two.

2.3 Graphlet-based methods and measures

Now that the definitions for graphlets and their orbits are established one can compute the graphlet degrees of nodes, layers or any other entities $\boldsymbol{\gamma}$ in multilayer networks. After this step, one can in general use multilayer graphlets in a very similar way as ordinary graphlets, by simply substituting multilayer graphlet degrees in place of single-layer graphlet degrees in any graphlet-based methods. For example, the graphlet degree vectors [16] and/or graphlet correlation matrices [17] can be used as an input for supervised or unsupervised learning tasks or they can be visualised and interpreted in the context of the input multilayer network data.

Consider for example a task of computing distances between networks. An existing statistic called the graphlet correlation distance (GCD) [17] is calculated based on graphlet correlation matrices, which are simply matrices regardless of if the orbits in question are single-layer or multilayer ones. The calculation of GCD is then exactly the same once graphlet degree

vectors have been found in multilayer networks (the procedure for calculating GCD shown in Supplementary Materials section H). Besides GCD, there are also other graphlet/orbit-based measures that can similarly be generalized to multilayer networks. These include the graphlet degree distribution agreement (GDDA) [16], the relative graphlet frequency distance (RGFD) [29], Netdis [18] (requires the definition of multilayer ego-networks), and NetEmd [14]. In the next section, GCD is taken as the multilayer distance measure of choice, and in Supplementary Materials section J, multilayer NetEmd (section J.1) and multilayer GDDA (section J.2) are investigated in more detail.

3 Comparing multiplex networks using graphlets

The framework for multilayer network graphlets encompasses a large number of use cases and types of networks. This comes at a cost of the framework being relatively abstract. We will next focus on the special case of graphlets in node-aligned single-aspect multiplex networks [2] in order to illustrate how the general multilayer network graphlet framework can be applied. This significantly simplifies many of the notions and serves as a concrete example for the various concepts we have introduced. To further illustrate the applicability of the graphlet framework, we use it to conduct a case study classifying multiplex network models in an unsupervised way using graphlet distance measures. Additionally, in Supplementary Materials section M, we present a case study for real-world multiplex protein-protein interaction networks where they are matched with the multiplex network models using different graphlet distance measures.

Multiplex networks are a type of multilayer networks that are diagonally coupled and where each layer shares at least one node with another layer [2]. In other words, all interlayer edges are between a node on some layer and the same node on another layer (i.e. have the form $((v, \boldsymbol{\alpha}), (v, \boldsymbol{\beta}))$) and there are no layers where the set of nodes is completely disjoint from every other layer. A node can be connected to many or even all of its counterparts on the other layers; it is only required that every interlayer edge has the same node on both ends, but there is no requirement that a node should have only one interlayer edge connected to it. In this section, we focus only on single-aspect multiplex networks where each layer contains the same set of nodes and every node is connected to all of its counterparts in other layers ($(v, \boldsymbol{\alpha}) \in V_M, (v, \boldsymbol{\beta}) \in V_M \implies ((v, \boldsymbol{\alpha}), (v, \boldsymbol{\beta})) \in E_M$). Such networks are called node-aligned with categorical couplings [2]. For simplicity we refer to them as multiplex networks in this section.

3.1 Graphlets in multiplex networks

We can enumerate multiplex graphlets by finding the isomorphism classes of all possible connected multiplex networks of desired size. Unlike in general multilayer networks, in multiplex networks we only need to consider different variations of intralayer edge configurations and the combinations of these intralayer networks, because by the definition of multiplex networks the same nodes are present in all of the layers and the interlayer edges are connecting the layers fully symmetrically. Figure 4 presents all multiplex graphlets with two nodes and two layers or three nodes and two layers when the isomorphism allows the permutation of both node and layer labels ($p = \{0, 1\}$). Graphlets with four nodes and two layers are

illustrated in Supplementary Materials section C. The two notions of connectivity discussed before (underlying network is connected, aggregated network is connected) are equivalent in our multiplex networks, since every node is connected to its counterparts in the other layers [4, 2]. Note that if we didn't allow the permutation of both node and layer labels in the isomorphism, the number of graphlets of given size would depend on the number of distinct nodes (if $p \in \{\emptyset, \{1\}\}$) and layers (if $p \in \{\emptyset, \{0\}\}$) in our network. In real-world applications this could correspond to a case where the nodes/layers contain some semantics that we want to preserve in our analysis [12].

Figure 4 also illustrates all the node orbits within the graphlets. The automorphism orbits are numbered starting from zero (we skip the arbitrary single-node graphlet and orbit), so in total there are 21 different automorphism orbits in the graphlets in Figure 4. When finding the graphlet degree distributions within a larger network, each orbit corresponds to one distribution. The node orbits of graphlets with four nodes and two layers are shown in Supplementary Materials section C. The number of node orbits grows quickly as the number of nodes and layers grows; Table 1 lists the number of orbits in graphlets with up to four nodes and three layers when node and layer labels are allowed to be permuted and when only node labels are allowed to be permuted (in the latter case, every graphlet is assumed to contain the same set of layer labels as the number of graphlets would grow with the set of possible layer labels).

As previously explained, we can consider layer and node-layer orbits in addition to node orbits. However, we limit our attention to node orbits and their graphlet degree distributions, and in the following text *orbit* refers to node orbit and *graphlet degree* refers to node graphlet degree, unless specified otherwise.

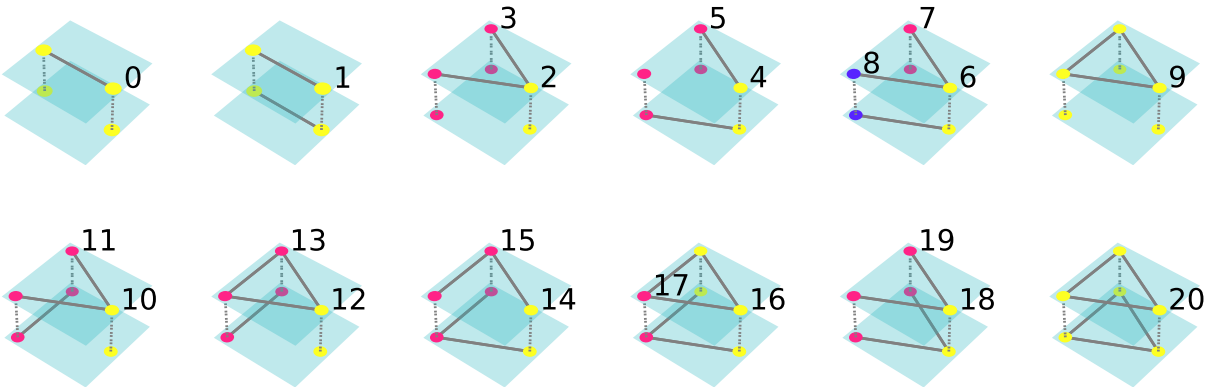


Figure 4: Graphlets and their automorphism orbits computed for nodes when nodes and layers are allowed to be permuted $Orb_{\{0,1\}}(u)$ for single-aspect multiplex networks with two layers and up to three nodes (single-node graphlet omitted). The orbits are numbered from 0 to 20. Within each graphlet, nodes colored with the same color belong to the same orbit.

3.2 Dependency equations for multiplex networks

We define an automatic process to construct equations that encode dependencies between orbit counts of different node orbits in multiplex networks. Each equation (for a pair of distinct orbits $x_1 \neq x_2$) has the form

$$\binom{C_{x_1}}{c_1}\binom{C_{x_2}-b}{c_2} = a_1 C_{y_1} + \ldots + a_k C_{y_k}\,, \tag{5}$$

where the left side of the equation represents all possible combinations of two orbits x_1 and x_2 with respect to a node, and the right side represents all the possible orbits y_j these combinations could generate. In other words, for each way two orbits can be combined, there must be some larger orbit that matches this combination (and so this larger orbit is one of $y_1, y_2, ..., y_k$). C_{x_i} and C_{y_j} are the counts of orbits x_i and y_j, respectively, c_1 and c_2 are the numbers of times orbits x_1 and x_2 are included in the combination, respectively, b is the number of times orbit x_2 is included in orbit x_1, and a_j is the number of ways a node can touch orbit y_j when it touches the combined orbits x_1 and x_2. We can also combine multiple instances of a single orbit, which results in an equation of slightly different form:

$$\binom{C_{x_1}}{c_1} = a_1 C_{y_1} + \ldots + a_k C_{y_k}\,, \tag{6}$$

where the elements are the same as before, except now only orbit x_1 is included c_1 times in the combination.

The process of constructing the equations is essentially the process of determining the orbits y_j and the coefficients a_j. The discovery of the orbits y_j that contribute to the counts of orbits x_i can be divided into three phases: combining the orbits, merging nodes and adding links. When combining orbits o_1 and o_2 (which can correspond to different orbits x_1 and x_2, as in Equation 5, or to the same orbit x_1, as in Equation 6) residing in graphlets g_1 and g_2 respectively, one of the nodes representative of orbit o_1 and one of the nodes representative of orbit o_2 are merged into a single node v_0 which will be connected to all the node-layers the two merged nodes were connected to. Otherwise the connections between nodes will remain unchanged. If layer labels are allowed to be permuted in the isomorphism, all possible mergings with respect to layer combinations are done (for example, if in g_1 there are layers 1 and 2 and in g_2 there are also layers 1 and 2, there's two possible mergings: matching layers 1 to 1 and 2 to 2, and matching layer 1 of one graphlet to layer 2 of the other graphlet and vice versa.) If layer labels are not allowed to be permuted, then the layers where the two nodes are found should match.

In the second phase, nodes originating from different graphlets in the resulting networks of the first phase can be merged while the network has at least $max(n_1, n_2)+1$ nodes, where n_1 and n_2 are the numbers of nodes in graphlets g_1 and g_2, respectively. For the merge to be allowed, the two nodes to be merged need to have (possible) edges connecting them to v_0 on exactly the same layers.

In the final phase, links can be added between node-layers that belonged to different graphlets (i.e. one belonged to g_1 and the other to g_2), since v_0 will still touch both orbits o_1 and o_2 in the resulting network. All the possible edge combinations should be added to all the obtained networks from the previous two phases. Since in our multiplex networks the interlayer edges are already specified, we can only add intralayer edges in this phase.

The orbits y_j can then be obtained by checking in which orbit node v_0 resides in each of the resulting networks. The coefficients a_j for the orbit counts C_{y_j} are obtained by computing the number of ways in which the graphlets g_1 and g_2 can be embedded into the resulting network such that v_0 touches orbit y_j. The subtrahend b in the left side of the equation is determined by calculating how many times a node in orbit o_1 touches orbit o_2 assuming graphlet g_1 has more nodes than g_2.

For example, if we combine two orbit 0s in multiplex networks with two layers (see Figure 4), the edges can be either in the same layer, resulting in orbit 2, or in different layers, resulting in orbit 4, when both node and layer labels are allowed to be permuted. The wedge-end nodes in these graphlets can be connected either in neither of the layers, only one of the layers, or both layers. Therefore, combining two orbit 0s can also result in orbits 9, 10, 11, 12 and 14, and the dependencies can be expressed with the equation $\binom{C_0}{2} = C_2 + C_4 + C_9 + C_{10} + C_{11} + C_{12} + C_{14}$, where C_i denotes the graphlet degree of a node with respect to orbit i. The dependency equations for up to 4-node graphlets in multiplex networks with 2 layers are listed in Supplementary Materials section D, and a computer program that produces these equations for any choice of parameters is provided as part of our multiplex graphlet analysis pipeline [27].

We could combine more than two orbits to create more complex dependency equations with the same process. However, these equations can be derived from the equations where only two orbits are combined (considering also that the orbit can be combined with itself), as shown in Supplementary Materials section F. Therefore, for the goal of discovering which equations are independent it is enough to consider equations where two orbits are combined.

A similar process applied here for multiplex networks could be constructed for general multilayer networks as well. However, there are further considerations that need to be taken into account in the general case. When merging two nodes in the first phase, we need to make sure that they appear on the same layers and have the same connectivity to their other instances on other layers, considering also all layer permutations where the aforementioned is true if layer labels are allowed to be permuted. When merging nodes in the second phase, we need to make sure they also appear on the same layers and have the same interlayer connectivity to their other instances, and also have the same connectivity to v_0. In the third phase, we must add all possible interlayer edges in addition to intralayer edges. In the general case, we can also consider any aspect or combination of aspects in addition to nodes for which to construct the equations.

3.3 Reducing multiplex orbit counts based on dependencies between them

Once we have generated a set of orbit dependency equations, we can discover which equations in the set are independent of the others. For each independent equation, we can calculate the value of one of the orbit count variables appearing in that equation based on the others. Therefore, for each independent equation, one orbit count is "redundant" and can be removed from the graphlet degree vectors of nodes in a network, to obtain reduced graphlet degree vectors. The process of finding these independent equations is described in detail in Supplementary Materials section G.

Table 1 describes the number of orbits (and therefore the lengths of unreduced graphlet

degree vectors) and the number of independent equations for graphlets up to four nodes and three layers, when both node and layer labels and only node labels are allowed to be permuted. For example, if we construct graphlet degree vectors of nodes in a network for orbits of graphlets with two layers and up to four nodes and $p = \{0, 1\}$, each vector will contain $2+19+391 = 412$ orbit counts. We can then reduce this vector to $412-(0+3+36) = 373$ orbit counts by removing one orbit count for each independent equation. As the number of nodes and layers grows, the number of orbits rapidly increases and the relative number of independent equations diminishes.

Finding a set of independent equations among a set of equations is not a process dependent on the multiplexity of the graphlets. Thus, the orbit count reduction method can also be applied to more general multilayer dependency equations, as long as such equations are first constructed.

Table 1: The number of all possible node orbits and independent orbit count equations for multiplex graphlets up to four nodes and three layers, when orbits are defined using either node-layer isomorphism or node isomorphism (in the latter case, each graphlet is assumed to contain the same set of layer labels, otherwise the number of graphlets and orbits would be infinite).

	Orbits						**Independent equations**					
Isomorphism type	Node-layer			Node			Node-layer			Node		
Nodes / Layers	*2*	*3*	*4*	*2*	*3*	*4*	*2*	*3*	*4*	*2*	*3*	*4*
1	1	3	11	1	3	11	0	1	3	0	1	3
2	2	19	391	3	33	751	0	3	36	0	6	91
3	3	67	8121	7	273	45311	0	6	193	0	28	1827

3.4 Multiplex network models

To assess the performance of the different distance measures and multilayer orbit definition approaches, we apply them to three single-aspect test sets of multiplex networks. In the first test set, we compare different random multiplex network models, some of which contain relevant edge correlations and overlaps between layers. This set consists of the eight models "BA-ind" – "WS", presented below. Between all networks, the average intralayer degrees of nodes are kept approximately constant. In the second set, we have the same models, but now we increase the average intralayer degree of networks for each model in a steadily progressing fashion, such that for every model there is the same mixture of different average intralayer degrees. In the third test set, we compare network groups which have different graphlets *purposefully inserted* into them, to mimic a real-world situation where some graphlets are enriched in the networks, for example because they are important for the function of the network or there is a particular mechanism creating such structures. This insertion method is described in "Graphlet insertion" subsection.

In the first and second set, we have eight different models, translating into eight different classes of networks. In the third set, we fix the number of classes to be five (different class means different graphlets inserted into the networks). In each set, we have 30 networks

per class (the first and second set thus have $30 \times 8 = 240$ and the third has $30 \times 5 = 150$ networks in total). Each network has 1000 nodes and 3 layers. In the first set, for most of the networks, the average intralayer degree of nodes is approximately equal to 4 (the average intralayer degree is approximately equal to $2m$, where m is the parameter of the Barabási-Albert (BA) networks. We set $m = 2$ in the first set.). In the second set, the average intralayer degrees progress as 2, 4, 6, 8, 10, and 12 ($m = 1, 2, 3, 4, 5, 6$, respectively), such that for each model there are five networks for each average intralayer degree. In the third set, the average intralayer degree of nodes is 4 in every network.

Independent Barabási-Albert models (BA-ind)

A multiplex network is constructed by generating a Barabási-Albert random network [31] for each layer independently of each other. First, a (complete) seed network with m nodes is created, and after that each new node is attached to m existing nodes with probabilities proportional to the degrees of the nodes. Each layer will have a power law degree distribution, but the degrees of a node between layers do not correlate and there is little overlap in the edges between layers. The average intralayer degree of a node is approximately $2m$. The node names are randomized for each layer separately and they do not follow the time when the nodes are introduced to the network.

Interdependent Barabási-Albert models (BA-dep)

Otherwise the same model as above, but new nodes are attached to the existing nodes with probabilities proportional to the sum of the degrees of a node over all the layers [32]. The model produces high interlayer degree correlations, but the overlap of edges between layers remains quite low.

Independent configuration models (Conf-ind)

Each layer of the multiplex network is constructed as a configuration model random network [1] independently of each other. The degree distributions of BA-ind networks are used to generate the Conf-ind networks. The model introduces low interlayer degree correlations and edge overlaps.

Interdependent configuration models (Conf-dep)

For each type of edge overlap (including having an edge only on a single layer), a configuration model [1] is generated. The final multiplex network is constructed by adding the edges from each configuration model to the layers corresponding to that overlap. The generated network approximately matches the degree correlations and edge overlaps of the network used as the basis. The multiplex degree distributions and edge overlaps of the BA-dep networks were used to generate the Conf-dep networks.

Zero-overlap Erdős-Rényi networks (ER-0)

Each layer is constructed as a separate Erdős-Rényi random network [33] with the restriction that an edge cannot exist in multiple layers. The model produces networks with zero edge

overlap and interlayer degree correlations close to zero. The average intralayer degree is $2m$, as in the Barabási-Albert networks, and when the network is aggregated into a single layer, the average degree is $2m$ times the number of layers (since there is no overlap between layers).

Overlapping Erdős-Rényi networks (ER-20)

Similarly as in the Conf-dep model, each type of overlap is generated as a regular Erdős-Rényi random network with the restriction that an edge can exist in at most one "overlap layer/network". The final multiplex network is then constructed by adding links of one of the networks to all the layers, links of another layer to all but one of the layers and so forth such that for each combination of layers we use the links of one generated network. With this model we generate networks with equal edge densities to ER-0 in the aggregated networks (average degree in an aggregated network is $2m$ times the number of layers), but with 20 % of the edges overlapping between every pair of layers. Because of the overlap, the average intralayer degree is therefore higher than $2m$.

Random geometric graphs with shared node location (GEO)

In the soft geometric random graph model [34], each node is randomly assigned a position in the unit square. Nodes within a threshold distance r are connected by an edge with probability e^{-d}, where d is their Euclidean distance. For the multiplex network, the nodes are positioned in the same locations in all the layers, but the edges between the nodes are added independently in each layer. The model produces networks with high edge overlap and interlayer degree correlations. The threshold distance is set to $r = \sqrt{\frac{2.2 \times m}{\pi \times (n-1)}}$, which produces average intralayer degrees of approximately $2m$.

Watts-Strogatz models with same initial lattice (WS)

The Watts-Strogatz model [35] starts with a ring and connects each node to its $2m$ nearest neighbors in all the layers. Then for each layer edges are rewired independently with probability $p = 0.3$. The model generates networks with high edge overlap, but with low interlayer degree correlations. The average intralayer degree is $2m$.

Graphlet insertion

To insert a graphlet with n nodes and l layers into a network, we randomly sample k instances of all possible combinations of n nodes and l layers in the network such that no two instances overlap for more than one node if they share at least one layer. We then change the edge configurations at those locations to match the inserted graphlet. We then add or remove edges outside those locations as needed to match the average intralayer degree on each layer with that of the original network, by either choosing random edges to be deleted (if there are more edges than in the original network) or by choosing two random nodes and adding an edge between them (if there are fewer edges than in the original network).

To construct a test set of networks with graphlet insertions, we repeat the following three steps five times: (1) We generate 30 layer-independent Erdős-Rényi (ER) multiplex

networks with three layers, 1000 nodes and average intralayer degree 4, (2) we randomly pick 20 different graphlets with $n = 4$ nodes and $l = 2$ layers from all such graphlets, (3) for each network in the ER networks, we insert $k = 3$ instances of each of the chosen graphlets into that network. As a result, we get five sets of 30 networks with different inserted graphlets. In Supplementary Materials section I, the graphlet insertion method is repeated for graphlets with 3 nodes and 3 layers.

3.5 Evaluation of measures

The GCDs for the network test sets were computed using graphlets with one, two and three layers. For the one-layer and two-layer cases, the orbits were counted for up to 3- and 4-node graphlets, and for the 3-layer case, the orbits were counted for only up to 3-node graphlets. The number of nodes here denotes the maximum number of nodes: for example, for four nodes and two layers, all graphlets with two layers and two, three, or four nodes were included. The orbits were computed for all the nodes in the networks using node-layer isomorphism. For the calculation of the one-layer orbits, the networks were layer-aggregated and treated as if they were single-layer networks to model the typical analysis procedure when ordinary graphs are used. For the 2-layer measures, the orbit counts were computed for each of the 2-layer combinations in the networks and the counts from all of the combinations were summed together to obtain the final orbit counts. The test set networks had three layers, so there was only one 3-layer combination to consider for the 3-layer graphlets. For each pair of number of nodes and number of layers, the distances were computed both including and excluding the redundant orbits.

The performance of each measure is evaluated by the area under the precision-recall curve (AUPR). Precision is defined as the fraction of true positives out of all positives and recall is the fraction of true positives and the sum of true positives and false negatives. Here, true positives are pairs of networks generated using the same model that have a distance smaller than the threshold ϵ at which the precision and recall values are evaluated. False positives are pairs of networks generated from different models but have a distance smaller than ϵ, and false negatives are the pairs of networks generated from the same model but have a distance larger than ϵ.

3.6 Comparison results

For the measures we use the following naming convention. The first number denotes the number of layers and the second is the maximum number of nodes in the graphlets. 'R' in the end denotes that redundant orbits have been removed. For example GCD-2-4R denotes GCD computed using orbits of multiplex graphlets with two layers and up to four nodes excluding the redundant orbits.

For comparison purposes, we also calculated GCDs following an alternative orbit definition GCD-DPK (named here after Dimitrova, Petrovski, Kocarev) [19] based on the implementation the authors have provided [36]. In the implementation, graphlets of two and three nodes are taken into account, so the performance is similar but not identical to our graphlets with three layers and up to three nodes. In general, our method produces different orbits than the alternative method as shown in Supplementary Materials section A.

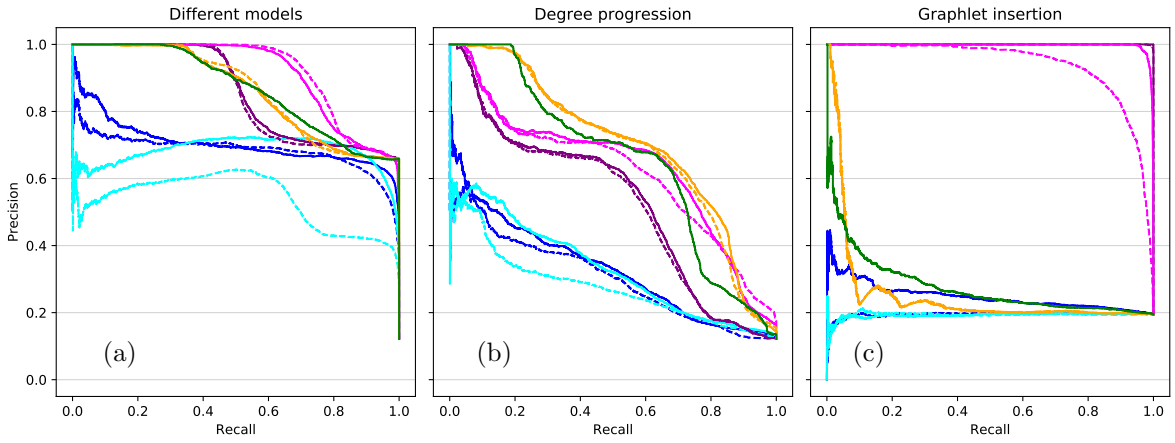
Comparison results for NetEmd and GDDA distances are shown in Supplementary Materials section J, and they support the conclusions drawn from the GCD results.

3.6.1 Precision-recall curves

We plot the precision-recall curves and AUPRs of all three multiplex network test sets in Figure 5. When separating different models with constant average intralayer degrees (Figure 5 (a)), GCD-2-3R has the highest AUPR, and all the multiplex graphlet methods perform better than the single-layer graphlet methods GCD-1-4(R) and GCD-1-3(R), showing that considering multiplex structure is indeed important when constructing graphlet-based distance measures for multiplex networks. When each model includes networks with progressing average intralayer degrees (Figure 5 (b)), the separation task is much more difficult. However, multilayer measures still perform much better than single-layer ones and GCD-3-3 has the highest AUPR. When separating graphlet insertion networks (Figure 5 (c)), GCD-2-4 achieves perfect separation. There is a clear difference between the performance of GCD-2-4(R) and GCD-2-3(R) compared to the other methods (and the other multilayer methods perform comparably to the single-layer methods). The difference is explained by the fact that the inserted graphlets had two layers and four nodes, which GCD-2-4(R) specifically finds in the networks, and that they therefore also contain some two-layer-three-node graphlets as subnetworks. When looking to separate real-world networks which are expected to contain different kinds of graphlets, one then has to choose a distance measure that contains the expected graphlets. The precision-recall curves illustrate that 1) methods created specifically with multiplex isomorphisms in mind are necessary for handling multiplex networks, and 2) just applying any multiplex method in an unsupervised context may not be good enough, instead the method should be adapted to whatever kind of graphlets the data contains (or is expected to contain).

Removing redundant orbits from the orbit counts slightly increases or decreases the AUPR in the two- and three-layer measures. On the other hand, there is a mostly performance-reducing effect with the single-layer measures, which might be explained by the smaller number of single-layer orbits compared to two- and three-layer orbits. The GCD-3-3(R) curves are close to the GCD-DPK curves, with GCD-3-3(R) performing slightly better in networks with degree progression, and slightly worse in constant degree and graphlet insertion networks.

For single-layer GCDs going from GCD-1-3 to GCD-1-4 increases the AUPR, whereas for the two-layer GCD increasing the graphlet size from three to four nodes decreases the performance except in the graphlet insertion networks. This could be explained by the fact that there are only few single-layer-three-node orbits and the number of orbits increases much faster as the number of nodes is increased when there are more layers (see Table 1). Thus, a greater proportion of the orbits in the two-layer four-node case are more likely to be completely non-existent or have a lot of counts close to zero especially if the networks are not very dense. For these orbits the ranks of the nodes used to compute the correlations between orbits can appear quite random. More elaboration on the orbit counts, causes for performance differences, and selecting the appropriate number of nodes and layers is presented in Supplementary Materials section L.



Distance (GCD)	1-4	1-4R	1-3	1-3R	2-4	2-4R	2-3	2-3R	3-3	3-3R	DPK
AUPR (a)	0.7050	0.6848	0.6816	0.5360	0.8568	0.8531	0.9150	**0.9226**	0.8608	0.8629	0.8672
AUPR (b)	0.3309	0.3038	0.3304	0.2752	0.5305	0.5238	0.6370	0.6211	**0.7031**	0.6936	0.6539
AUPR (c)	0.2481	0.1938	0.1924	0.1930	**1.0000**	0.9999	0.9969	0.9395	0.2506	0.2535	0.2759

Figure 5: Precision-recall curves and AUPRs for different distance measures. The dashed lines depict the measures where redundant orbits have been removed. (a) Test set with eight different multiplex network models, all networks have similar approximate intralayer degrees. (b) Test set with eight different multiplex network models, and for each model the average intralayer degrees progressed from 2 to 12 in steps of 2. (c) Test set with graphlets with four nodes and two layers inserted.

3.6.2 Multidimensional scaling embeddings

To see how well the different models are grouped by different distance measures, the networks are embedded into 3-dimensional space using the multidimensional scaling method (MDS) [37] which preserves the different distances as well as possible. Figure 6 shows the embeddings for the different multiplex models with constant average intralayer degree; the embeddings for progressing intralayer degree and graphlet insertion networks are shown in Supplementary Materials section I. All the multiplex methods separate the network classes somewhat clearly, except for the BA-ind and Conf-ind, and BA-dep and Conf-dep models. These models have pairwise matched edge overlaps between layers, i.e. their multiplex structure is very similar, which may explain the difficulty in separating them.

3.6.3 Pairwise AUPRs

Figure 7 illustrates the pairwise AUPR values when clustering networks from different random network models with average intralayer degree progression. The pairwise AUPRs for the random networks with constant intralayer degrees and graphlet insertion networks are shown in Supplementary Materials section I. The single-layer measures GCD-1-3 and GCD-1-4 perform worse than the multilayer measures overall, and as expected they are especially unsuited for cases, such as the two ER models, that are designed to be statistically indistinguishable at the aggregated network level. All the measures have trouble distinguishing BA-ind, BA-dep, Conf-ind, and Conf-dep from one another, especially in BA-ind–Conf-ind and BA-dep–Conf-dep pairs. None of the multilayer methods is the best in every situa-

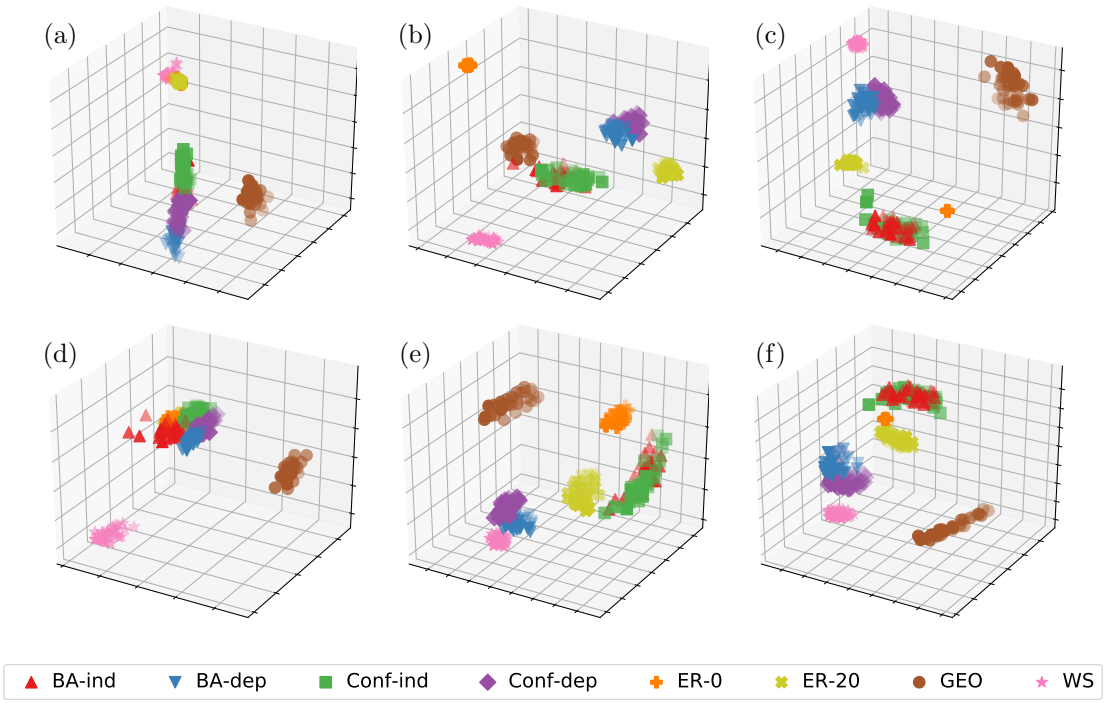


Figure 6: Networks from different multiplex models with constant average intralayer degrees embedded into 3-dimensional space using MDS while preserving different distance measures. (a) GCD-1-3, (b) GCD-2-3, (c) GCD-3-3, (d) GCD-1-4, (e) GCD-2-4, (f) GCD-DPK. The multiplex measures generally separate the the models better than the single-layer models do. Separating BA-ind from Conf-ind and BA-dep from Conf-dep is difficult, while the other models form clearly separate clusters with the multiplex measures.

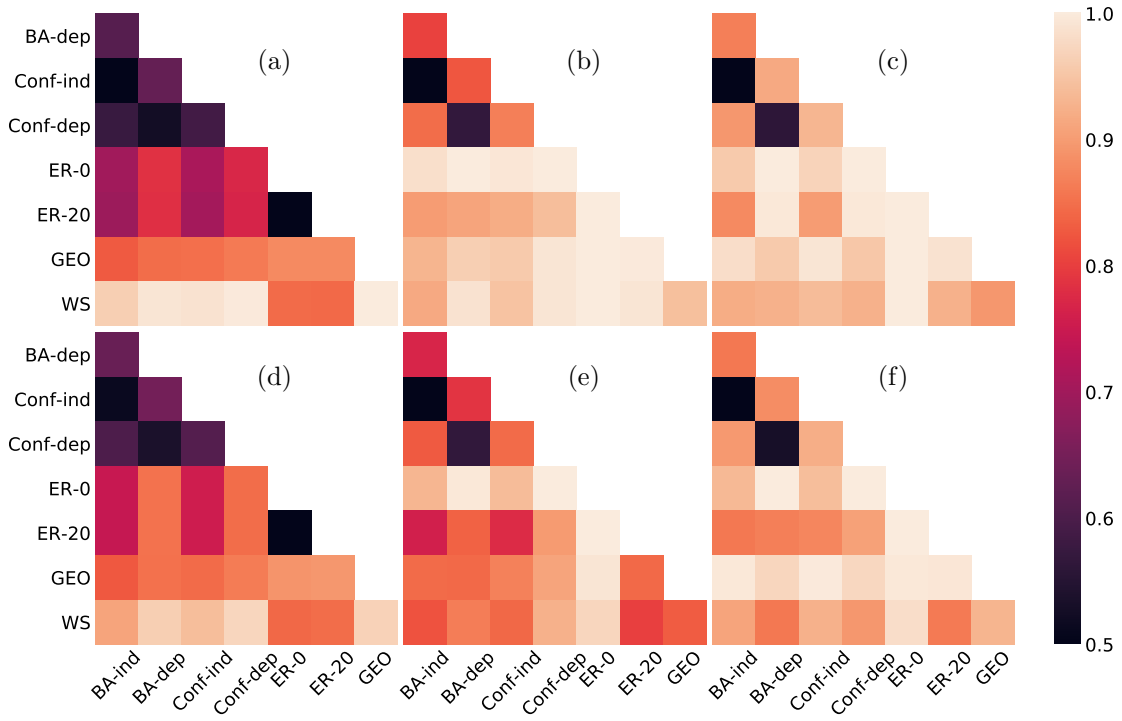


Figure 7: Pairwise AUPRs when separating networks with average intralayer degree progression from different models using different distance measures. (a) GCD-1-3, (b) GCD-2-3, (c) GCD-3-3, (d) GCD-1-4, (e) GCD-2-4, (f) GCD-DPK.

tion, however GCD-3-3 seems to have similar but overall slightly better performance than GCD-DPK in most pairs. The results highlight that multilayer methods are better than single-layer ones also in pairwise separation tasks.

4 Discussion

To answer the need for graphlet-based tools for investigating the structure of multilayer networks, we have created a systematic and principled framework for multilayer network graphlet analysis, starting from the definition of isomorphism and automorphism orbits in multilayer networks. The framework can be used with any kind of multilayer network with any number of aspects. We have illustrated the usefulness of the framework with test sets of multiplex networks, showing that 1) multilayer graphlet correlation distance performs considerably better in grouping networks from different multiplex random models than single-layer graphlet correlation distance, and that 2) when there is graphlet structure in the networks, choosing the right multilayer graphlet size in the correlation distance measure is highly important, and that 3) the number of redundancy equations is relatively small compared to the number of orbits in multiplex networks and using them does not lead to significant improvements in the unsupervised prediction tasks we constructed.

We have presented the graphlet analysis pipeline in the light of comparing a set of net-

works to one another and in a general format, leaving for example the choice of isomorphism to the reader. Naturally, the choices made in the pipeline (Figure 1) affect the end results, and it might not be immediately clear what the correct ones for a given situation are. Especially when there are graphlets of a specific size in the network(s) we are investigating and we include graphlets of the wrong size in our graphlet-based measures, the results can be devastating in unsupervised learning tasks (see Figure 5 (c)).

Once one obtains the graphlet degree vectors or the graphlet correlation matrix, they can be used as feature vectors/matrices in a variety of ways [38, 39, 40, 41, 42, 43]. The analysis presented here on clustering multiplex networks using node orbits and node-layer isomorphism barely scratches the surface of the possibilities given by multilayer graphlet analysis. Clearly, other types of multilayer networks can be analyzed with different isomorphisms. Additionally, the possibility to define orbits for layers, and combinations of layers, in addition to nodes could result in interesting new ways of analyzing multilayered systems. Further, in addition to unsupervised learning problems (such as clustering networks), supervised learning models can be constructed to e.g. predict which group a network or an element of a network belongs in based on the graphlet degrees. Even though such methods have been previously applied to single-layer networks [44, 45, 46, 47], there is a clear avenue for further research in the application of machine learning to multilayer graphlets. Presumably, the problem of choosing the correct set of graphlets is less severe in supervised learning in which the method should be able to find the important orbits based on the training data.

In this article, we applied the framework to multiplex networks generated from different multiplex network random models. Such synthetic networks are convenient for establishing a "ground truth" for evaluation of network measures, but they do not necessarily reflect well the properties of real-world multiplex and especially multilayer networks. Since there are already multiple well-labeled multilayer/plex data sets available (e.g. [48]), the usefulness of the multilayer graphlet framework should be established on them. Further, the exact choice of the models will probably reflect on the results greatly. For example, apart from the graphlet insertion method, our models did not include complex interlayer relations, which probably would highlight the importance of larger graphlets.

The generalization of single-layer graphlet and orbit concepts and methods to multilayer networks massively expands the possibilities of network analysis with different types of data, relationships, and hierarchies present in real-world networked systems. Starting from the mathematical foundation of multilayer network automorphisms provides a solid theoretical basis for multilayer orbits, on which advanced concepts and theory can be built in the future. The freedom gained in the leap from single-layer to multilayer networks enables more accurate representation and structural analysis of network-like systems, opening the door for a multitude of applications in a wide variety of fields where multilayer networks appear, ranging from natural sciences and engineering to social sciences and humanities.

Funding

This work was supported by the Academy of Finland project ECANET [320781 to MK].

References

[1] M. Newman, *Networks: an introduction*. Oxford university press, 2010.

[2] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.

[3] S. Boccaletti, G. Bianconi, R. Criado, C. I. Del Genio, J. Gómez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, no. 1, pp. 1–122, 2014.

[4] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *science*, vol. 328, no. 5980, pp. 876–878, 2010.

[5] M. Magnani, O. Hanteer, R. Interdonato, L. Rossi, and A. Tagarelli, "Community detection in multiplex networks," *arXiv:1910.07646*, 2019.

[6] X. Huang, D. Chen, T. Ren, and D. Wang, "A survey of community detection methods in multilayer networks," *Data Mining and Knowledge Discovery*, vol. 35, no. 1, pp. 1–45, 2021.

[7] M. Salehi, R. Sharma, M. Marzolla, M. Magnani, P. Siyari, and D. Montesi, "Spreading processes in multilayer networks," *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 2, pp. 65–83, 2015.

[8] M. De Domenico, A. Solé-Ribalta, E. Omodei, S. Gómez, and A. Arenas, "Centrality in interconnected multilayer networks," *arXiv preprint arXiv:1311.2906*, 2013.

[9] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas, "Mathematical formulation of multilayer networks," *Physical Review X*, vol. 3, no. 4, p. 041022, 2013.

[10] F. Battiston, V. Nicosia, and V. Latora, "Structural measures for multiplex networks," *Physical Review E*, vol. 89, no. 3, p. 032804, 2014.

[11] E. Cozzo, M. Kivelä, M. De Domenico, A. Solé-Ribalta, A. Arenas, S. Gómez, M. A. Porter, and Y. Moreno, "Structure of triadic relations in multiplex networks," *New Journal of Physics*, vol. 17, no. 7, p. 073029, 2015.

[12] M. Kivelä and M. A. Porter, "Isomorphisms in multilayer networks," *arXiv preprint arXiv:1506.00508*, 2015.

[13] Ö. N. Yaveroğlu, T. Milenković, and N. Pržulj, "Proper evaluation of alignment-free network comparison methods," *Bioinformatics*, vol. 31, no. 16, pp. 2697–2704, 2015.

[14] A. E. Wegner, L. Ospina-Forero, R. E. Gaunt, C. M. Deane, and G. Reinert, "Identifying networks with common organizational principles," *arXiv preprint arXiv:1704.00387*, 2017.

[15] S. Aliakbary, S. Motallebi, S. Rashidian, J. Habibi, and A. Movaghar, "Distance metric learning for complex networks: Towards size-independent comparison of network structures," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 2, p. 023111, 2015.

[16] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, 2007.

[17] Ö. N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, and N. Pržulj, "Revealing the hidden language of complex networks," *Scientific reports*, vol. 4, 2014.

[18] W. Ali, T. Rito, G. Reinert, F. Sun, and C. M. Deane, "Alignment-free protein interaction network comparison," *Bioinformatics*, vol. 30, no. 17, pp. i430–i437, 2014.

[19] T. Dimitrova, K. Petrovski, and L. Kocarev, "Graphlets in multiplex networks," *Scientific Reports*, vol. 10, no. 1, pp. 1–13, 2020.

[20] S. Jiao, Z. Xue, X. Chen, and Y. Xu, "Sampling graphlets of multi-layer networks: A restricted random walk approach," *arXiv preprint arXiv:2001.07136*, 2020.

[21] F. Battiston, V. Nicosia, M. Chavez, and V. Latora, "Multilayer motif analysis of brain networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 4, p. 047404, 2017.

[22] H. D. Boekhout, W. A. Kosters, and F. W. Takes, "Efficiently counting complex multilayer temporal motifs in large-scale networks," *Computational Social Networks*, vol. 6, no. 1, pp. 1–34, 2019.

[23] J. Enright and K. Meeks, "Counting small subgraphs in multi-layer networks," *arXiv preprint arXiv:1710.08758*, 2017.

[24] F. W. Takes, W. A. Kosters, B. Witte, and E. M. Heemskerk, "Multiplex network motifs as building blocks of corporate networks," *Applied network science*, vol. 3, no. 1, pp. 1–22, 2018.

[25] Y. Ren, A. Sarkar, A. Ay, A. Dobra, and T. Kahveci, "Finding conserved patterns in multilayer networks," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 97–102, 2019.

[26] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 601–610, 2017.

[27] S. Sallmen, T. Nurmi, and M. Kivelä. `https://github.com/bolozna/multiplex-graphlet-analysis`, version: 86cbd5c.

[28] P. M. Cohn, *Algebra Volume 1*. Wiley, 2 ed., 1982.

[29] N. Pržulj, D. G. Corneil, and I. Jurisica, "Modeling interactome: scale-free or geometric?," *Bioinformatics*, vol. 20, no. 18, pp. 3508–3515, 2004.

[30] A. Sarajlić, N. Malod-Dognin, Ö. N. Yaveroğlu, and N. Pržulj, "Graphlet-based characterization of directed networks," *Scientific reports*, vol. 6, 2016.

[31] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[32] J. Y. Kim and K.-I. Goh, "Coevolution and correlated multiplexity in multiplex networks," *Physical review letters*, vol. 111, no. 5, p. 058702, 2013.

[33] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[34] M. D. Penrose *et al.*, "Connectivity of soft random geometric graphs," *The Annals of Applied Probability*, vol. 26, no. 2, pp. 986–1028, 2016.

[35] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, p. 440, 1998.

[36] K. Petrovski. `https://github.com/K5rovski/graphlet_pyframework`, version: 49f7d89.

[37] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC press, 2000.

[38] M. Tantardini, F. Ieva, L. Tajoli, and C. Piccardi, "Comparing methods for comparing networks," *Scientific reports*, vol. 9, no. 1, pp. 1–19, 2019.

[39] T. Milenković, W. L. Ng, W. Hayes, and N. Pržulj, "Optimal network alignment with graphlet degree vectors," *Cancer informatics*, vol. 9, pp. CIN–S4744, 2010.

[40] T. Milenković and N. Pržulj, "Uncovering biological network function via graphlet degree signatures," *Cancer informatics*, vol. 6, pp. CIN–S680, 2008.

[41] V. Vijayan, V. Saraph, and T. Milenković, "Magna++: Maximizing accuracy in global network alignment via both node and edge conservation," *Bioinformatics*, vol. 31, no. 14, pp. 2409–2411, 2015.

[42] T. Lyu, Y. Zhang, and Y. Zhang, "Enhancing the network embedding quality with structural similarity," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 147–156, 2017.

[43] W. Hayes, K. Sun, and N. Pržulj, "Graphlet-based measures are suitable for biological network comparison," *Bioinformatics*, vol. 29, no. 4, pp. 483–491, 2013.

[44] Y. Zhang, Y. Chen, and T. Hu, "Panda: Prioritization of autism-genes using network-based deep-learning approach," *Genetic epidemiology*, vol. 44, no. 4, pp. 382–394, 2020.

[45] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein, "Improving graph neural network expressivity via subgraph isomorphism counting," *arXiv preprint arXiv:2006.09252*, 2020.

[46] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *Artificial intelligence and statistics*, pp. 488–495, PMLR, 2009.

[47] Q. Li and T. Milenkovic, "Improving supervised prediction of aging-related genes via dynamic network analysis," *arXiv preprint arXiv:2005.03659*, 2020.

[48] M. E. Dickison, M. Magnani, and L. Rossi, *Multilayer Social Networks*. Cambridge University Press, 2016.

Supplementary Materials: Graphlets in multilayer networks

Sallamari Sallmen* Tarmo Nurmi*† Mikko Kivelä*

June 7, 2022

A Comparison of orbit definition methods

We compare our method to that of Dimitrova et al. [1], and show that the two methods do not yield the same orbits by a counter-example.

Consider the network in Figure 1. Intuitively, the "roles" of nodes 1 and 3 are similar in the network, and the role of 2 is different from them. Now, $\boldsymbol{\zeta} : 1 \leftrightarrow 3, a \leftrightarrow d, b \leftrightarrow c$ is a node-layer automorphism. According to our definition, $Orb_{\{0,1\}}(1) = \{1, 3\} = Orb_{\{0,1\}}(3)$ and $Orb_{\{0,1\}}(2) = \{2\}$, corresponding to the intuition.

However, using the definitions of Dimitrova et al., the sub-orbits of the nodes are 1: $3_{ab.ad.cd}$, 2: $3_{ab.cd.ad}$, 3: $3_{ad.cd.ab}$, i.e. they are all different. The first reduction method gives orbits 1: $3_{2.2.2}$, 2: $3_{2.2.2}$, 3: $3_{2.2.2}$, i.e. they are all the same. The second reduction method gives orbits 1: $3_{2x.2y.2z}$, 2: $3_{2x.2y.2z}$, 3: $3_{2x.2y.2z}$, i.e. they are all the same. Therefore, the orbits of this method are different from the orbits defined by our method, and the underlying intuition is also different.

A.1 Number of graphlet degree distributions and the size of the network

With our method, for graphlets of specific size, the number of graphlet degree distributions does not depend on the size of the network when all aspects of the network are allowed to be permuted in the isomorphism. That is, the number of orbits for a specific graphlet size is fixed. With the Dimitrova et al. method, the number of orbits very quickly explodes if not reduced, scaling exponentially with the number of layers [1]. The size of the graphlet correlation matrix (GCM) is orbits times orbits, so computing it quickly becomes prohibitively expensive. With the Dimitrova et al. method, in our test networks with 3 layers, the number of orbits (dimension of GCM) was 280. In networks with 4 layers, it was 2160. In networks with 5 layers, it was 16864. With 5 layers, each GCM (16864 times 16864) already consumed 7.3-7.4 GB of disk space when pickle-serialized from a numpy array, the total coming up to

*Department of Computer Science, Aalto University School of Science, P.O. Box 15400, FI-00076, Finland
†tarmo.nurmi@aalto.fi

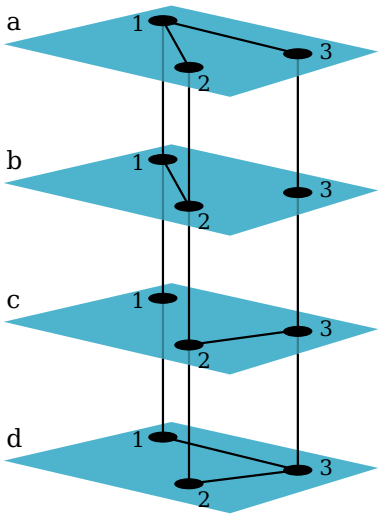


Figure 1: Example multiplex network with $V = \{1, 2, 3\}, L_1 = \{a, b, c, d\}$.

1.1 TB for the whole 150-network set. This made it practically impossible to calculate the distance measures with Dimitrova et al. orbits for networks with more than 5 layers.

On the other hand, we calculated the precision-recall curves (Figure 2), MDS embeddings (Figure 3), and pairwise AUPRs (Figure 4) for a set of 3-node-3-layer graphlet insertion networks with 10 layers (10 graphlets inserted, 350 each) with our method. In Figures 3 and 4 we use the following convention for subfigures: **top left:** GCD-1-3; **top middle:** GCD-2-3; **top right:** GCD-3-3; **bottom left:** GCD-1-4; **bottom middle:** GCD-2-4. The number of different orbits is the same as with any other number of layers, and applicability of the method is only limited by computational time required to find the graphlet degree vectors.

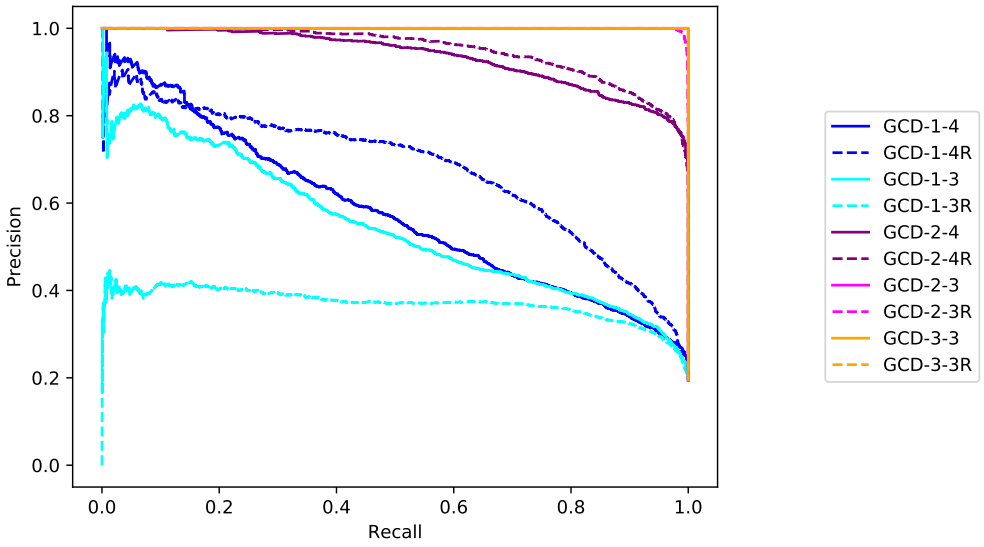


Figure 2: 10-layer 3-node-3-layer graphlet insertion precision-recall

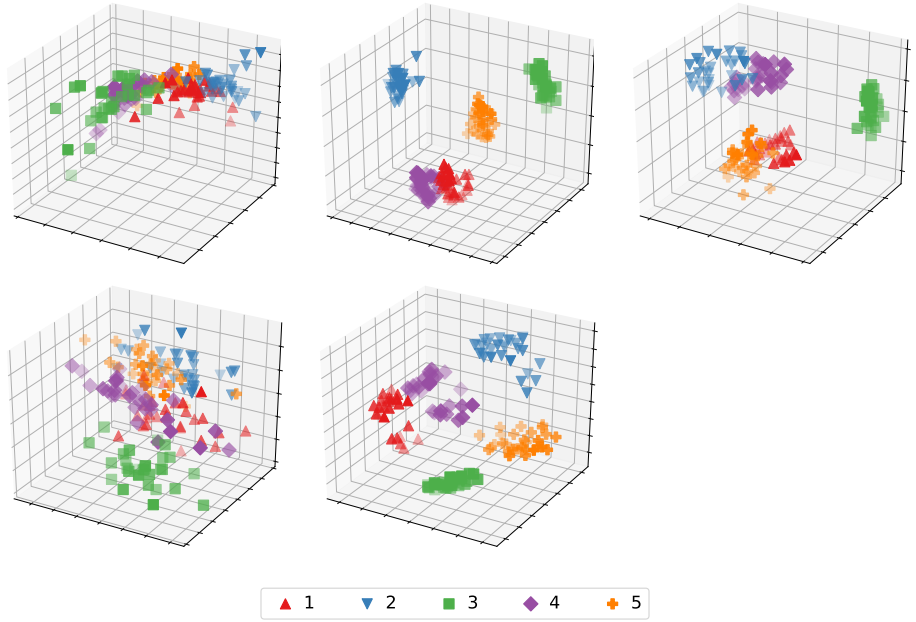


Figure 3: 10-layer 3-node-3-layer graphlet insertion MDS

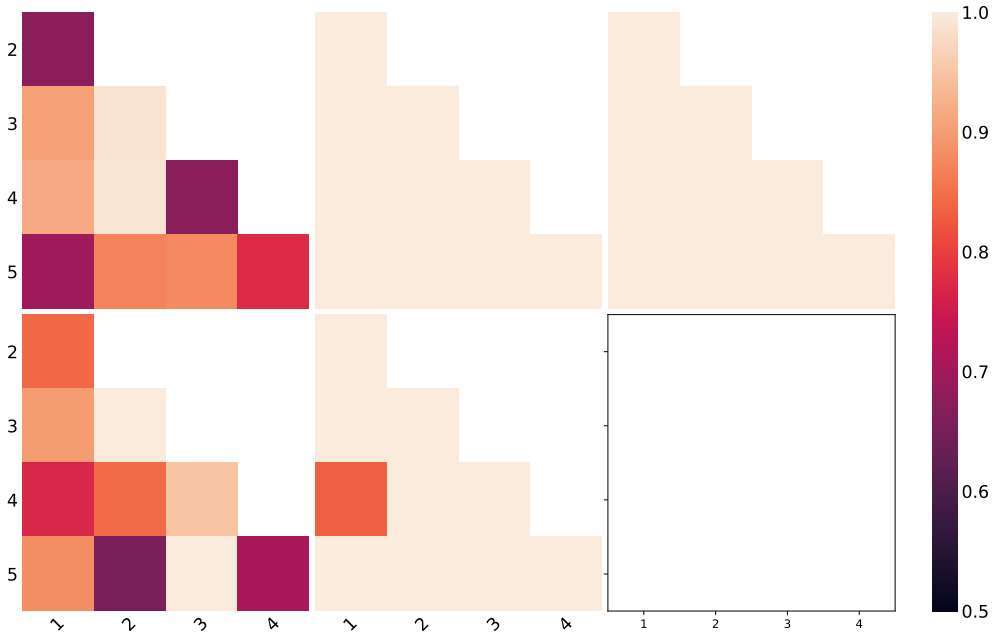


Figure 4: 10-layer 3-node-3-layer graphlet insertion pairwise AUPRs

B Detailed proof for multilayer automorphism orbits induce equivalence classes

Let $\sim$ be a relation with $\boldsymbol{\gamma} \sim \boldsymbol{\delta} \Leftrightarrow \boldsymbol{\delta} \in Orb_p(\boldsymbol{\gamma})$. Now, we need to prove that $\sim$ is an equivalence relation, i.e. that it is reflexive, symmetric, and transitive. The identity permutation is always an automorphism, and therefore $\exists\ \boldsymbol{\zeta}$ such that $\boldsymbol{\zeta}(\boldsymbol{\gamma}) = \boldsymbol{\gamma}\ \forall\ \boldsymbol{\gamma}$. Then, $\boldsymbol{\gamma} \sim \boldsymbol{\gamma}$ and $\sim$ is reflexive. If $\boldsymbol{\gamma} \sim \boldsymbol{\delta}$, $\exists\ \boldsymbol{\zeta}$ such that $\boldsymbol{\delta} = \boldsymbol{\zeta}(\boldsymbol{\gamma})$. Every automorphism $\boldsymbol{\zeta}$ has an inverse $\boldsymbol{\zeta}^{-1}$ that's also an automorphism, because $\boldsymbol{\zeta}$ maps a multilayer network to itself and we can therefore reverse the relabeling done in $\boldsymbol{\zeta}$ to also map the network to itself. Then, $\boldsymbol{\delta} = \boldsymbol{\zeta}(\boldsymbol{\gamma}) \Leftrightarrow \boldsymbol{\zeta}^{-1}(\boldsymbol{\delta}) = \boldsymbol{\zeta}^{-1}(\boldsymbol{\zeta}(\boldsymbol{\gamma})) \Leftrightarrow \boldsymbol{\zeta}^{-1}(\boldsymbol{\delta}) = \boldsymbol{\gamma}$ which means that $\boldsymbol{\gamma} \in Orb_p(\boldsymbol{\delta})$ and $\boldsymbol{\delta} \sim \boldsymbol{\gamma}$. Therefore, $\boldsymbol{\gamma} \sim \boldsymbol{\delta} \Leftrightarrow \boldsymbol{\delta} \sim \boldsymbol{\gamma}$ and $\sim$ is symmetric. If we combine two automorphisms, the result is also an automorphism, because each automorphism maps a multilayer network back to itself. If $\boldsymbol{\gamma}_1 \sim \boldsymbol{\gamma}_2$ and $\boldsymbol{\gamma}_2 \sim \boldsymbol{\gamma}_3$, then $\exists\ \boldsymbol{\zeta}_1$ such that $\boldsymbol{\zeta}_1(\boldsymbol{\gamma}_1) = \boldsymbol{\gamma}_2$ and $\exists\ \boldsymbol{\zeta}_2$ such that $\boldsymbol{\zeta}_2(\boldsymbol{\gamma}_2) = \boldsymbol{\gamma}_3$. Now, $\boldsymbol{\zeta}_2\boldsymbol{\zeta}_1$ is also an automorphism, and $\boldsymbol{\zeta}_2\boldsymbol{\zeta}_1(\boldsymbol{\gamma}_1) = \boldsymbol{\zeta}_2(\boldsymbol{\zeta}_1(\boldsymbol{\gamma}_1)) = \boldsymbol{\gamma}_3$, so $\boldsymbol{\gamma}_3 \in Orb_p(\boldsymbol{\gamma}_1)$ and $\boldsymbol{\gamma}_1 \sim \boldsymbol{\gamma}_3$. Therefore, $\boldsymbol{\gamma}_1 \sim \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_2 \sim \boldsymbol{\gamma}_3 \implies \boldsymbol{\gamma}_1 \sim \boldsymbol{\gamma}_3$ and $\sim$ is transitive. □

C Orbits for graphlets with two layers and four nodes

We list the multiplex graphlets with two layers and four nodes and their orbits ($p = \{0, 1\}$) in Figures 5–7.

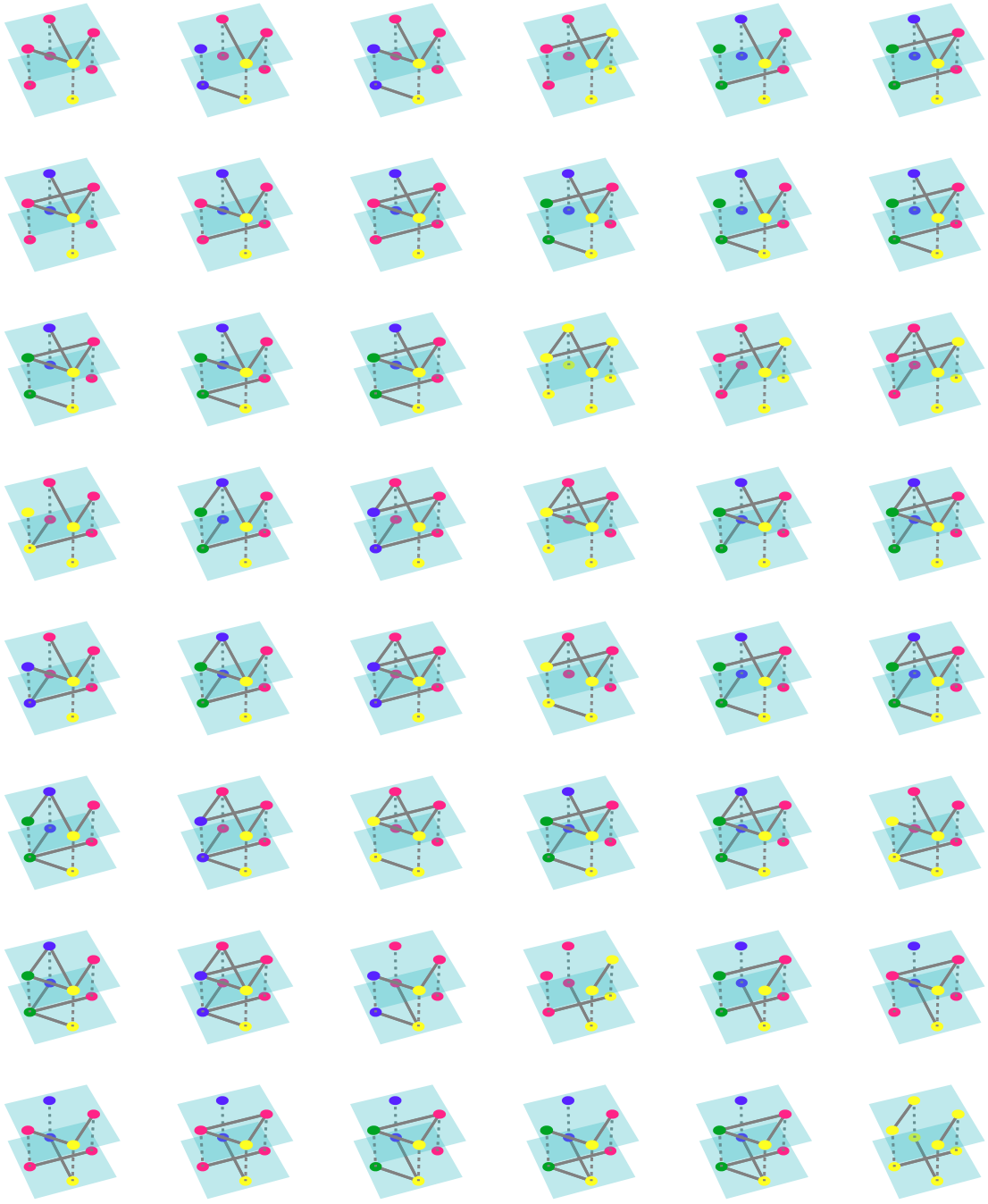


Figure 5: 2-layer-4-node multiplex graphlets 1-48 out of 137 and their orbits for nodes using node-layer isomorphism $Orb_{\{0,1\}}(u)$. Within a graphlet nodes colored with the same color belong to the same orbit.

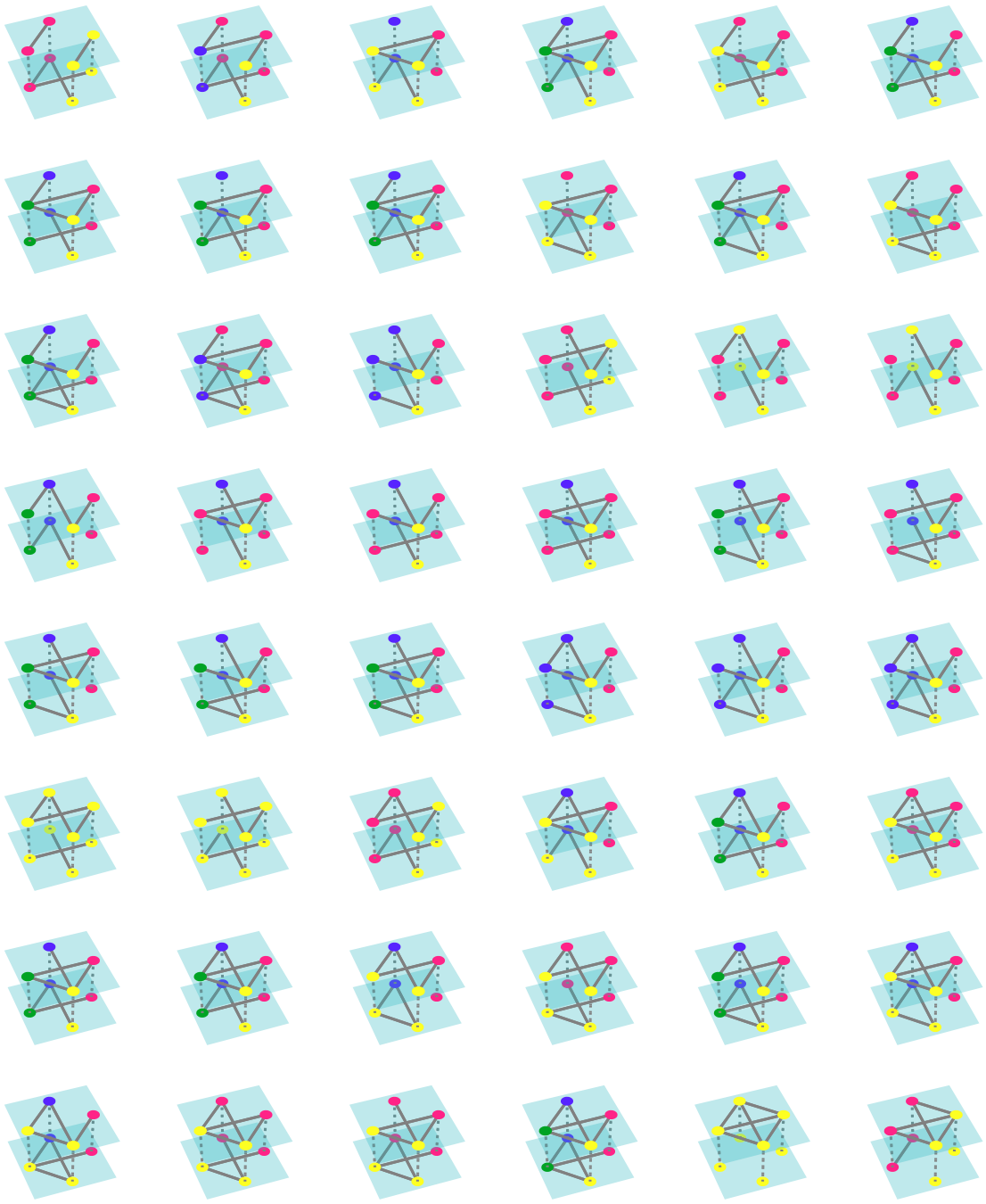


Figure 6: 2-layer-4-node multiplex graphlets 49-96 out of 137 and their orbits for nodes using node-layer isomorphism $Orb_{\{0,1\}}(u)$. Within a graphlet nodes colored with the same color belong to the same orbit.

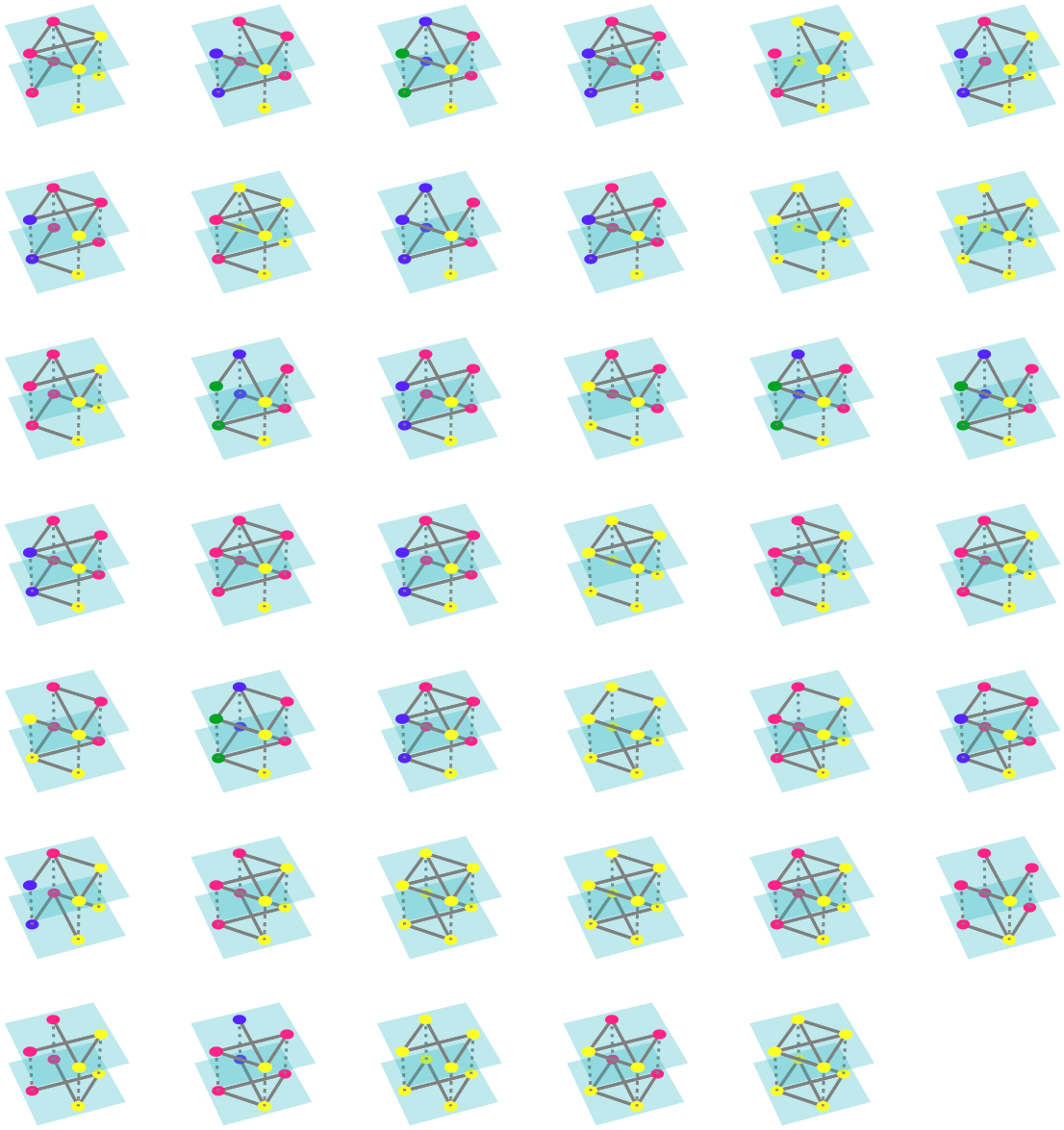


Figure 7: 2-layer-4-node multiplex graphlets 97-137 out of 137 and their orbits for nodes using node-layer isomorphism $Orb_{\{0,1\}}(u)$. Within a graphlet nodes colored with the same color belong to the same orbit.

D Orbit equations for up to 4-node graphlets in multiplex networks with 2 layers

The first three equations are obtained when combining two graphlets that produce graphlets with 3 nodes. The graphlets combined in the rest of the equations produce graphlets with 4 nodes. With the algorithm described in section G, we obtain that equations 14 and 15 can be derived from the rest of the equations.

$$\binom{C_0}{2} = C_2 + C_4 + C_9 + C_{10} + C_{11} + C_{12} + C_{14} \quad (1)$$

$$\binom{C_1}{2} = C_{17} + C_{18} + C_{20} \quad (2)$$

$$\binom{C_1}{1}\binom{C_0}{1} = C_6 + C_{13} + C_{15} + C_{16} \quad (3)$$

$$\binom{C_2}{1}\binom{C_0-2}{1} = 3C_{21} + C_{23} + 2C_{39} + 2C_{42} + 2C_{45} + C_{48} + C_{52} + C_{56} + C_{86} + C_{88} + C_{92} + C_{96} + C_{98} + C_{99} + C_{103} + C_{106} + C_{108} + C_{112} + C_{116} + C_{120} \quad (4)$$

$$\binom{C_3}{1}\binom{C_0-1}{1} = C_{29} + C_{32} + C_{40} + C_{43} + C_{49} + C_{53} + C_{61} + C_{65} + 2C_{72} + C_{73} + C_{74} + C_{75} + 2C_{78} + C_{80} + 2C_{87} + C_{89} + C_{90} + C_{93} + 2C_{97} + C_{100} + 2C_{107} + C_{109} + C_{110} + C_{113} + C_{117} + 2C_{124} + C_{126} + C_{127} + C_{130} + 2C_{134} + C_{136} \quad (5)$$

$$\binom{C_4}{1}\binom{C_0-2}{1} = 2C_{23} + C_{48} + C_{52} + C_{56} + C_{91} + C_{111} + 2C_{151} + 2C_{154} + 2C_{157} + C_{178} + C_{181} + C_{185} + C_{187} + C_{191} + C_{195} + C_{199} \quad (6)$$

$$\binom{C_5}{1}\binom{C_0-1}{1} = C_{31} + C_{51} + C_{55} + C_{73} + C_{74} + 2C_{77} + C_{79} + C_{89} + C_{90} + C_{109} + C_{110} + C_{126} + C_{127} + C_{145} + C_{152} + C_{155} + C_{161} + C_{165} + 2C_{172} + C_{173} + 2C_{179} + 2C_{180} + C_{182} + 2C_{186} + C_{188} + C_{193} + C_{197} + 2C_{204} + C_{206} + 2C_{210} + C_{212} \quad (7)$$

$$\binom{C_6}{1}\binom{C_0-1}{1} = 2C_{26} + C_{60} + C_{64} + C_{68} + C_{95} + C_{102} + C_{115} + C_{119} + 2C_{142} + C_{160} + C_{164} + C_{168} + C_{184} + C_{190} + C_{194} + C_{198} + 2C_{231} + 2C_{234} + 2C_{237} + 2C_{240} + 2C_{244} + C_{272} + C_{275} + C_{278} + C_{279} + C_{281} + C_{284} + C_{285} + C_{289} + C_{292} + C_{294} \quad (8)$$

$$\binom{C_7}{1}\binom{C_0-1}{1} = C_{35} + C_{75} + C_{79} + C_{80} + 2C_{83} + C_{93} + C_{100} + C_{113} + C_{117} + C_{130} + C_{136} + C_{147} + C_{173} + 2C_{175} + C_{182} + C_{188} + C_{193} + C_{197} + C_{206} + C_{212} + C_{232} + C_{235} + C_{241} + C_{243} + C_{248} + C_{252} + 2C_{273} + 2C_{276} + 2C_{290} + 2C_{299} + 2C_{302} \quad (9)$$

$$\binom{C_8}{1}\binom{C_0}{1} = C_{63} + C_{67} + C_{76} + C_{81} + C_{82} + C_{94} + C_{101} + C_{114} + C_{118} + C_{131} + C_{137} + C_{163} + C_{167} + C_{174} + C_{183} + C_{189} + C_{192} + C_{196} + C_{207} + C_{213} + C_{223} + C_{225} + C_{261} + C_{264} + C_{268} + C_{269} + C_{280} + C_{282} + C_{283} + C_{293} + C_{305} + C_{307} \quad (10)$$

$$\binom{C_9}{1}\binom{C_0-2}{1} = C_{39} + 2C_{86} + C_{88} + C_{91} + C_{92} + C_{151} + C_{178} + C_{181} + 3C_{312} + 2C_{313} + C_{314} + 2C_{315} + C_{317} + C_{318} + C_{320} + C_{321} + C_{324} + C_{327} + C_{329} + C_{332} \quad (11)$$

$$\binom{C_{10}}{1}\binom{C_0-2}{1} = C_{42} + C_{88} + 2C_{96} + C_{99} + C_{111} + C_{154} + C_{185} + C_{187} + C_{313} + 2C_{317} + C_{319} + C_{320} + 3C_{328} + 2C_{337} + C_{340} + C_{343} + C_{344} + C_{345} + C_{347} + C_{351} \quad (12)$$

$$\binom{C_{11}}{1}\binom{C_0-2}{1} = C_{48} + C_{52} + C_{91} + 2C_{98} + 2C_{106} + 2C_{108} + C_{111} + C_{112} + C_{116} + C_{178} + C_{185} + C_{191} + C_{195} + 2C_{314} + 2C_{318} + 2C_{319} + C_{321} + 2C_{327} + C_{329} + 2C_{338} + 2C_{343} + 2C_{344} + C_{345} + C_{347} + 2C_{348} + 2C_{355} + C_{357} + C_{361} \quad (13)$$

$$\binom{C_{12}}{1}\binom{C_0-2}{1} = C_{45}+C_{92}+C_{99}+2C_{103}+C_{157}+C_{191}+C_{195}+C_{199}+C_{315}+C_{320}+2C_{324}+C_{337}+ \quad (14)$$
$$C_{338}+2C_{340}+C_{348}+C_{355}+C_{357}+C_{361}+3C_{367}+C_{369}$$

$$\binom{C_{13}}{1}\binom{C_0-1}{1} = C_{60}+C_{95}+C_{102}+2C_{123}+C_{125}+C_{128}+C_{129}+C_{160}+C_{194}+C_{198}+2C_{203}+C_{205}+ \quad (15)$$
$$C_{279}+C_{284}+2C_{316}+C_{322}+C_{323}+C_{325}+2C_{339}+C_{341}+C_{349}+C_{350}+2C_{354}+$$
$$C_{356}+C_{362}+2C_{372}+C_{373}+C_{374}+C_{375}+2C_{378}+C_{380}$$

$$\binom{C_{14}}{1}\binom{C_0-2}{1} = C_{56}+C_{112}+C_{116}+2C_{120}+C_{181}+C_{187}+C_{199}+C_{321}+C_{329}+2C_{332}+C_{345}+C_{347}+ \quad (16)$$
$$2C_{351}+C_{357}+C_{361}+2C_{369}$$

$$\binom{C_{15}}{1}\binom{C_0-1}{1} = C_{64}+C_{115}+C_{119}+C_{125}+C_{128}+2C_{133}+C_{135}+C_{164}+C_{184}+C_{190}+2C_{209}+C_{211}+ \quad (17)$$
$$C_{281}+C_{292}+C_{322}+C_{323}+2C_{330}+2C_{331}+C_{333}+2C_{346}+C_{349}+C_{350}+C_{352}+$$
$$C_{358}+C_{360}+C_{373}+C_{374}+2C_{377}+C_{379}+2C_{386}+C_{387}$$

$$\binom{C_{16}}{1}\binom{C_0-1}{1} = C_{68}+C_{129}+C_{135}+2C_{139}+C_{168}+C_{205}+C_{211}+2C_{215}+C_{272}+C_{275}+C_{278}+C_{285}+ \quad (18)$$
$$C_{289}+C_{294}+C_{325}+C_{333}+2C_{335}+C_{341}+C_{352}+C_{356}+C_{358}+C_{360}+C_{362}+2C_{364}+$$
$$2C_{365}+C_{375}+C_{379}+C_{380}+2C_{383}+C_{387}+2C_{389}$$

$$\binom{C_{17}}{1}\binom{C_0}{1} = C_{132}+C_{138}+C_{208}+C_{214}+C_{259}+C_{262}+C_{304}+C_{306}+C_{326}+C_{334}+C_{342}+C_{353}+ \quad (19)$$
$$C_{359}+C_{363}+C_{376}+C_{381}+C_{382}+C_{388}+C_{397}+C_{398}$$

$$\binom{C_{18}}{1}\binom{C_0}{1} = C_{105}+C_{122}+C_{202}+C_{218}+C_{247}+C_{251}+C_{255}+C_{288}+C_{297}+C_{395} \quad (20)$$

$$\binom{C_{19}}{1}\binom{C_0}{1} = C_{84}+C_{104}+C_{121}+C_{140}+C_{176}+C_{200}+C_{201}+C_{216}+C_{227}+C_{250}+C_{254}+C_{270}+ \quad (21)$$
$$C_{286}+C_{295}+C_{309}+C_{392}$$

$$\binom{C_{20}}{1}\binom{C_0}{1} = C_{265}+C_{298}+C_{301}+C_{308}+C_{368}+C_{370}+C_{371}+C_{384}+C_{390}+C_{399} \quad (22)$$

$$\binom{C_2}{1}\binom{C_1}{1} = C_{26}+C_{60}+C_{64}+C_{68}+C_{123}+C_{125}+C_{129}+C_{133}+C_{135}+C_{139} \quad (23)$$

$$\binom{C_3}{1}\binom{C_1}{1} = C_{36}+C_{46}+C_{57}+C_{69}+C_{76}+C_{81}+C_{84}+C_{94}+C_{101}+C_{104}+C_{114}+C_{118}+C_{121}+ \quad (24)$$
$$C_{131}+C_{137}+C_{140}$$

$$\binom{C_4}{1}\binom{C_1}{1} = C_{128}+C_{142}+C_{160}+C_{164}+C_{168}+C_{203}+C_{205}+C_{209}+C_{211}+C_{215} \quad (25)$$

$$\binom{C_5}{1}\binom{C_1}{1} = C_{59}+C_{82}+C_{148}+C_{158}+C_{169}+C_{174}+C_{176}+C_{183}+C_{189}+C_{192}+C_{196}+C_{200}+ \quad (26)$$
$$C_{201}+C_{207}+C_{213}+C_{216}$$

$$\binom{C_6}{1}\binom{C_1-1}{1} = C_{132}+C_{138}+C_{208}+C_{214}+2C_{218}+C_{247}+C_{251}+C_{255}+2C_{259}+2C_{262}+2C_{265}+ \quad (27)$$
$$C_{298}+C_{301}+C_{304}+C_{306}+C_{308}$$

$$\binom{C_7}{1}\binom{C_1}{1} = C_{221}+C_{238}+C_{245}+C_{256}+C_{268}+C_{269}+C_{270}+C_{280}+C_{282}+C_{283}+C_{286}+C_{293}+ \quad (28)$$
$$C_{295}+C_{305}+C_{307}+C_{309}$$

$$\binom{C_8}{1}\binom{C_1-1}{1} = C_{71}+2C_{85}+C_{171}+2C_{177}+C_{229}+C_{267}+C_{271}+2C_{274}+2C_{277}+C_{287}+2C_{291}+ \quad (29)$$
$$C_{296}+2C_{300}+2C_{303}+C_{310}$$

$$\binom{C_9}{1}\binom{C_1}{1} = C_{95} + C_{184} + C_{231} + C_{272} + C_{316} + C_{322} + C_{325} + C_{330} + C_{333} + C_{335} \quad (30)$$

$$\binom{C_{10}}{1}\binom{C_1}{1} = C_{119} + C_{194} + C_{234} + C_{275} + C_{331} + C_{350} + C_{354} + C_{356} + C_{360} + C_{364} \quad (31)$$

$$\binom{C_{11}}{1}\binom{C_1}{1} = C_{102} + C_{115} + C_{190} + C_{198} + C_{240} + C_{278} + C_{289} + C_{323} + C_{339} + C_{341} + C_{346} + C_{349} + C_{352} + C_{358} + C_{362} + C_{365} \quad (32)$$

$$\binom{C_{12}}{1}\binom{C_1}{1} = C_{237} + C_{279} + C_{281} + C_{285} + C_{372} + C_{373} + C_{375} + C_{377} + C_{379} + C_{383} \quad (33)$$

$$\binom{C_{13}}{1}\binom{C_1 - 1}{1} = 2C_{105} + C_{132} + C_{202} + C_{208} + C_{247} + C_{288} + C_{298} + 2C_{326} + 2C_{342} + C_{359} + C_{363} + 2C_{368} + C_{370} + C_{376} + C_{381} + C_{384} \quad (34)$$

$$\binom{C_{14}}{1}\binom{C_1}{1} = C_{244} + C_{284} + C_{292} + C_{294} + C_{374} + C_{378} + C_{380} + C_{386} + C_{387} + C_{389} \quad (35)$$

$$\binom{C_{15}}{1}\binom{C_1 - 1}{1} = 2C_{122} + C_{138} + C_{202} + C_{214} + C_{251} + C_{297} + C_{301} + 2C_{334} + 2C_{353} + C_{359} + C_{363} + C_{370} + 2C_{371} + C_{382} + C_{388} + C_{390} \quad (36)$$

$$\binom{C_{16}}{1}\binom{C_1 - 1}{1} = C_{255} + C_{288} + C_{297} + C_{304} + C_{306} + C_{308} + C_{376} + C_{381} + C_{382} + C_{384} + C_{388} + C_{390} + 2C_{395} + 2C_{397} + 2C_{398} + 2C_{399} \quad (37)$$

$$\binom{C_{17}}{1}\binom{C_1 - 2}{1} = 2C_{141} + 2C_{217} + C_{311} + 3C_{336} + 3C_{366} + 2C_{385} + 2C_{391} + C_{393} + C_{400} \quad (38)$$

$$\binom{C_{18}}{1}\binom{C_1 - 2}{1} = C_{141} + C_{217} + C_{311} + 2C_{393} + 3C_{401} + 2C_{405} + C_{409} \quad (39)$$

$$\binom{C_{19}}{1}\binom{C_1 - 1}{1} = C_{258} + C_{271} + C_{287} + C_{296} + C_{310} + 2C_{396} + C_{403} + C_{406} + 2C_{408} + 2C_{410} \quad (40)$$

$$\binom{C_{20}}{1}\binom{C_1 - 2}{1} = C_{311} + C_{385} + C_{391} + 2C_{400} + C_{405} + 2C_{409} + 3C_{411} \quad (41)$$

E Number of generated orbit equations

Table 1 shows the number of orbit equations generated by our method for graphlets with a specific number of layers and nodes when using node-layer isomorphism or node isomorphism. These set the upper bounds for how many *independent* equations can be found.

Table 1: Number of generated equations for graphlets with given number of layers and nodes with node-layer and node isomorphisms.

		Node-layer			Node		
	Nodes	2	3	4	2	3	4
Layers	1	0	1	3	0	1	3
	2	0	3	38	0	6	99
	3	0	6	201	0	28	1911

F Combining more than two orbits

In the main article, we have only discussed how to generate equations where two orbits are combined, even though one could form equations where three or more orbits are combined. However, these equations depend on the equations where two orbits are combined and can be derived from them. Thus, they cannot be used to remove redundant orbits from the graphlet degree vectors.

In general, the equations combining any number of orbits can be expressed in the following form

$$\binom{C_{x_1}}{c_1}\binom{C_{x_2}-b_2}{c_2}\cdots\binom{C_{x_l}-b_l}{c_l} \;=\; a_1 C_{y_1} + \ldots + a_k C_{y_k}\ . \tag{42}$$

In three-orbit equations, i.e. equations where three orbits are combined, all the combined orbits can be different, two of the orbits can be the same or all of them can be the same. When all the orbits are the same, the three-orbit equation $\binom{C_x}{3}$ can be derived from equation $\binom{C_x}{2}$ by multiplying both sides of the equation by $(C_x-2)/3$. Equation $\binom{C_{x_1}}{1}\binom{C_{x_2}-b}{2}$ can be derived by multiplying equation $\binom{C_{x_1}}{1}\binom{C_{x_2}-b}{1}$ by $(C_{x_2}-b-1)/2$, and equation $\binom{C_{x_1}}{1}\binom{C_{x_2}-b_2}{1}\binom{C_{x_3}-b_3}{1}$ by multiplying $\binom{C_{x_1}}{1}\binom{C_{x_2}-b_2}{1}$ by $C_{x_3}-b_3$. The left sides of the equations become of the desired form, and the right sides of the equations become of the form $a_1 C_{y_1}(C_x-b)+\ldots+a_k C_{y_k}(C_x-b)$. The terms $C_y(C_x-b)$ correspond to two-orbit equations that have been defined already and can be replaced by the right sides of the equations to get the three-orbit equation in the form of equation 42.

For the case $\binom{C_x}{3}$, there is only one equation it can be derived from, $\binom{C_x}{2}$. In addition to that equation, it will also depend on the equations emerging in the right side of the multiplied equation. For the case $\binom{C_{x_1}}{1}\binom{C_{x_2}-b}{2}$, there are two possible equations it can be derived from, equation $\binom{C_{x_1}}{1}\binom{C_{x_2}-b}{1}$ or equation $\binom{C_{x_2}}{2}$. When $b>0$, the equation is not in form 42, but

with some rearranging we obtain

$$\begin{aligned}
\binom{C_{x_2}}{2}\binom{C_{x_1}}{1} &= \frac{1}{2}C_{x_1}C_{x_2}(C_{x_2}-1) \\
&= \frac{1}{2}C_{x_1}C_{x_2}(C_{x_2}-b)+\frac{1}{2}(b-1)C_{x_1}C_{x_2} \\
&= \frac{1}{2}C_{x_1}(C_{x_2}-b-1)(C_{x_2}-b)+\frac{1}{2}(b+1)C_{x_1}(C_{x_2}-b) \\
&\quad +\frac{1}{2}(b-1)C_{x_1}(C_{x_2}-b)+\frac{1}{2}b(b-1)C_{x_1} \\
&= \binom{C_{x_1}}{1}\binom{C_{x_2}-b}{2}+b\binom{C_{x_1}}{1}\binom{C_{x_2}-b}{1}+\frac{1}{2}b(b-1)C_{x_1}\ .
\end{aligned}$$

Therefore, the equation $\binom{C_{x_1}}{1}\binom{C_{x_2}-b}{2}$ depends on equation $\binom{C_{x_1}}{1}\binom{C_{x_2}-b}{1}$ in addition to equation $\binom{C_{x_2}}{2}$, and the equations obtained by multiplying the right side of that equation by C_{x_1}. For the equation $\binom{C_{x_1}}{1}\binom{C_{x_2}-b_2}{1}\binom{C_{x_3}-b_3}{1}$, there are three possibilities.

Equations combining more than three orbits can be derived in a similar manner from equations combining fewer nodes. These equations will not affect the independency of two-orbit equations, and thus will not be discussed in more detail.

When inspecting which equations can be used to derive two-orbit equations, one can search for one of the two orbits from the right sides of the equations and multiply this equation by the other orbit count. Then the equation depends on all the other two-orbit equations emerging on the right side of the equation and the three-orbit equation appearing on the left side of the equation. The other option would be to search one of the orbit counts from the left sides of the equation and analogously multiply that equation by the other orbit count. The formed equation could then be divided by the third orbit count to achieve the desired form on the left side of the equation. However, on the right side of the equation there would be divisions by orbit counts for which we do not have defined equations.

G Finding independent equations

There are two-orbit equations that can be set independent for certain. The equations that contain orbit counts that do not exist in any other equation cannot be derived from the other equations, and thus are independent. Moreover, the equations where the orbit combination produces graphlets of size at most three nodes can be set independent. These equations do not depend on each other, since each resulting two-star will represent a different graphlet, and one can investigate which of the equations can be derived from these equations.

To discover a set of independent equations, a network is formed, where the equations represent the nodes and there is a directed edge from equation e_1 to equation e_2 if equation e_2 is required for the derivation of equation e_1. Next, a directed acyclic graph (DAG) is formed out of the strongly connected components of the network. The components are processed in the linearized order of the DAG starting from sinks and proceeding to sources. This way the equations will only depend on equations within the same component and equations in the earlier components of the linearized order.

The networks in sinks are first set independent. Within other components, an equation can be set dependent if the equations it has links to have already been determined either

independent or dependent. If the component forms a complete graph, one of the equations can be set dependent and the rest are set independent. If the component contains a three-orbit equation, it should be set dependent, since those equations are not explicitly generated.

If the component does not form a complete graph, one of the equations, e_0, in that component is selected and set dependent. If the component contains three-orbit equations, these should be selected first. Next, another equation, e_1, is selected. If it can be derived from the other equations in the component and equations in the earlier components without the first selected equation e_0, then the equation e_1 is also set dependent. Otherwise, e_1 is left in the component and another equation is selected as e_1. One continues this way selecting equations and determining whether they can be derived from the remaining equations to find the largest set of equations that can be set dependent in the component.

H Calculating graphlet correlation distance (GCD)

Consider for example a task of computing distances between networks based on their node-based orbits [2]. In this task, the ith graphlet degree of a node depicts the number of graphlets in the network where the node is in orbit i. These graphet degrees can be collected in a vector called the graphlet degree vector (GDV) [2]. One can reduce the number of orbits to be included in the computation of these measures by using dependency equations (demonstrated in the multiplex network comparison section). For each independent equation, one orbit can be ignored altogether, reducing the size of the GDVs. The graphlet correlation matrix of a network (GCM) [3] is obtained by computing Spearman's correlation coefficients between vectors containing the orbit counts for all the nodes in the network. The dimensions of the GCM are therefore *number of orbits* $\times$ *number of orbits*, enabling the comparison of networks of different sizes. The correlation coefficient cannot be computed if all the counts for a certain orbit are equal, therefore a 'dummy node' with all orbit counts equal to one is added to the set of GDVs. These matrices for different networks can then be compared with a statistic called the graphlet correlation distance (GCD) [3] which is obtained by taking the Euclidean distance between the upper triangles of the GCMs,

$$GCD(G,H) = \sqrt{\sum_{i=1}^{n_{orb}} \sum_{j=i+1}^{n_{orb}} (GCM_G(i,j) - GCM_H(i,j))^2}\,, \tag{43}$$

where n_{orb} is the number of orbits used to compute the GCMs. The value of GCD is dependent on the set of orbits we choose to include in the GCMs. One common way of choosing orbits is to select a maximum size for the graphlets and include the orbits within all graphlets of at most that size. In multilayer networks, as discussed before, the graphlet size can be defined by giving a size for each aspect. The more aspects there are, the more choices have to be made to define the set of orbits to be used in the measure.

I Additional figures for GCD

In the main article, the precision-recall curves were shown for the constant degree random models, degree progression random models, and 4-node-2-layer graphlet insertion networks, with GCD as the distance measure. Additionally, MDS embedding for the constant degree random models and pairwise AUPRs for the degree progression random models were shown, also with GCD. Here, we list the rest of the figures with GCD that were not included in the main article: pairwise AUPRs for the constant degree random models (Figure 8); MDS embedding for the degree progression random models (Figure 9); and MDS embedding (Figure 10) and pairwise AUPRs (Figure 11) for the 4-node-2-layer graphlet insertion networks.

Additionally, we run the analysis for graphlet insertion networks where 3-node-3-layer graphlets were inserted. The procedure is otherwise the same as in the 4-node-2-layer case, except now we only insert 10 different graphlets (instead of 20), because there are much fewer 3-node-3-layer graphlets than 4-node-2-layer graphlets. Three instances of each graphlet were inserted. The resulting precision-recall curves (Figure 12), MDS embedding (Figure 13) and pairwise AUPRs (Figure 14) show that, as expected, GCD-3-3 performs the best in this case.

In Figures 8, 9, 10, 11, 13, and 14 we use the following convention for subfigures: **top left:** GCD-1-3; **top middle:** GCD-2-3; **top right:** GCD-3-3; **bottom left:** GCD-1-4; **bottom middle:** GCD-2-4; **bottom right:** GCD-DPK.

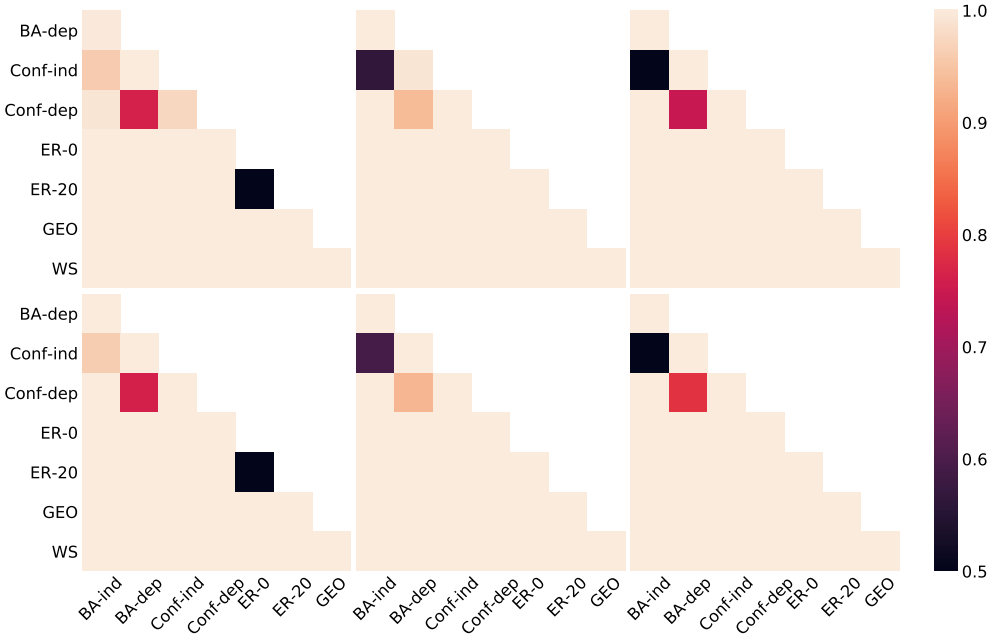


Figure 8: Constant degree pairwise AUPRs

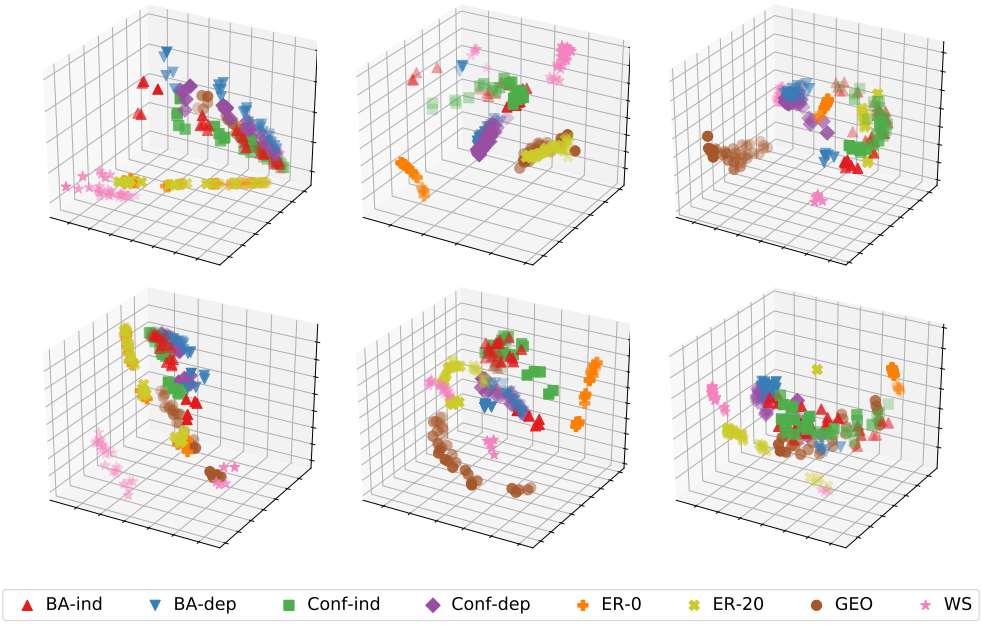


Figure 9: Degree progression MDS

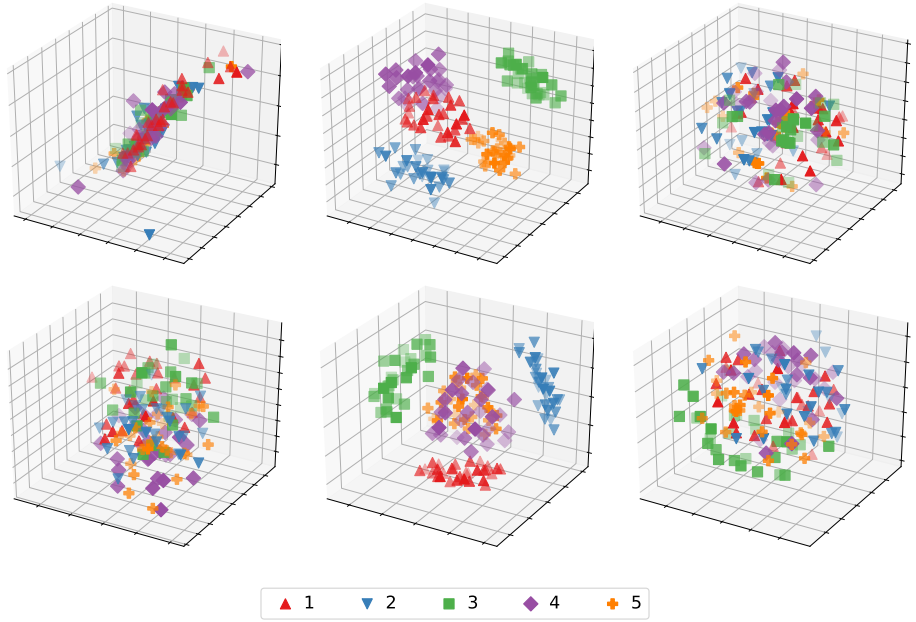


Figure 10: 4-node-2-layer graphlet insertion MDS

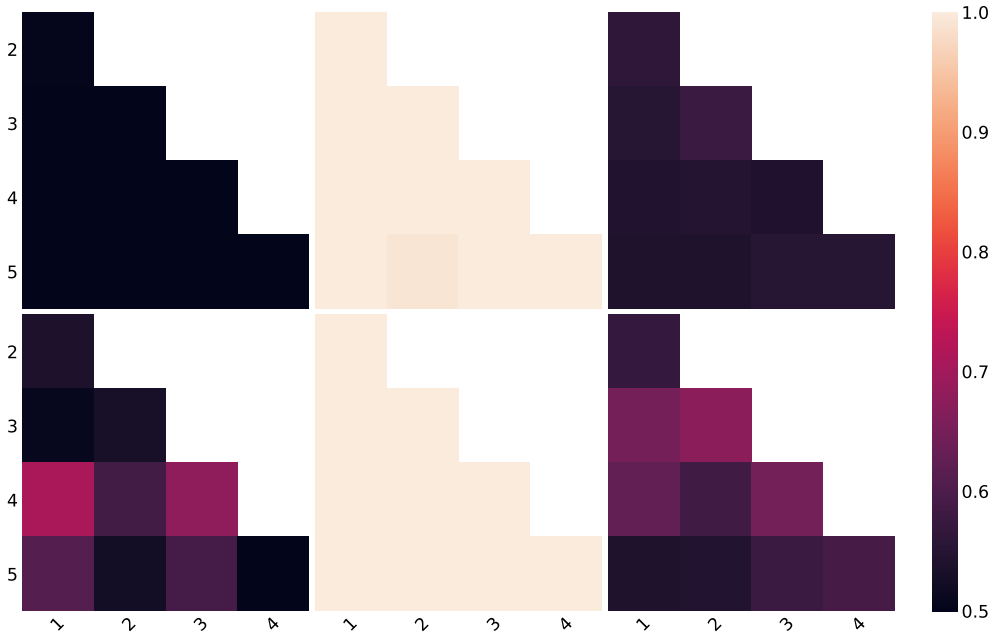


Figure 11: 4-node-2-layer graphlet insertion pairwise AUPRs

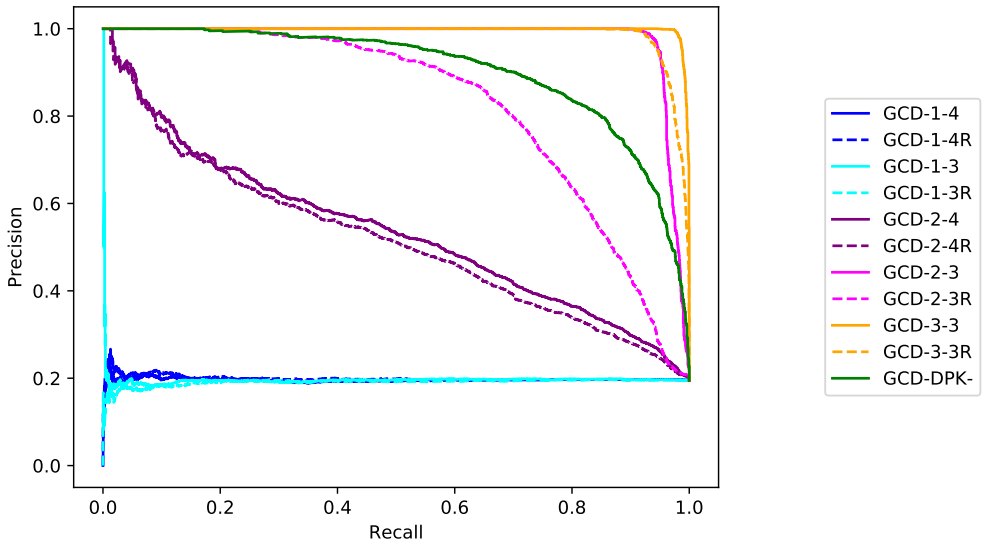


Figure 12: 3-node-3-layer graphlet insertion precision-recall

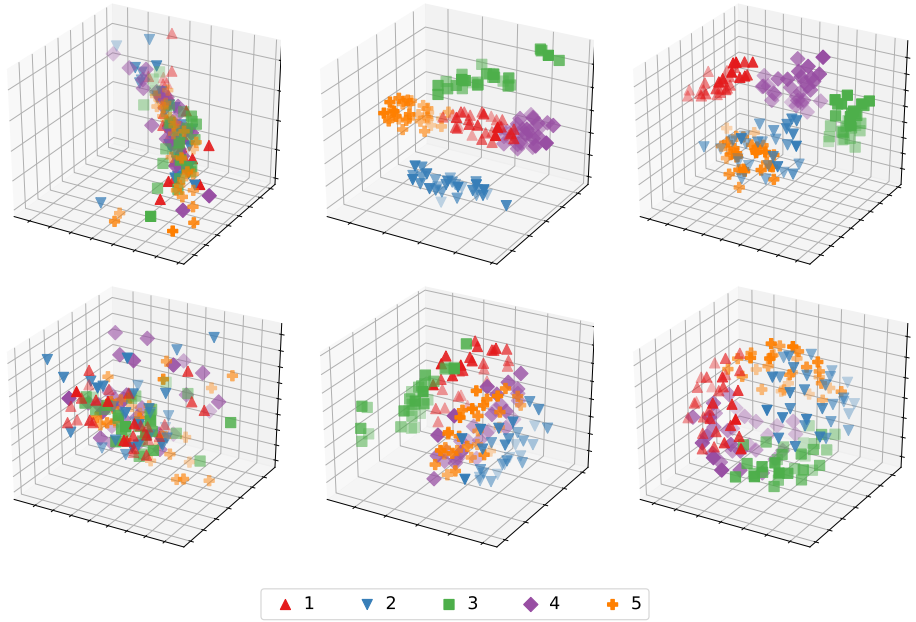


Figure 13: 3-node-3-layer graphlet insertion MDS

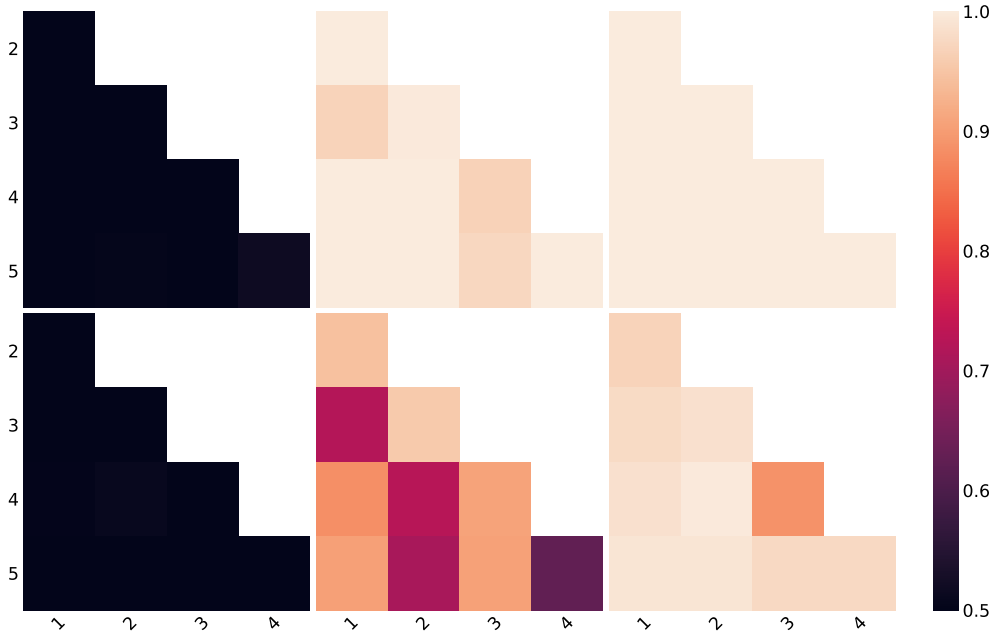


Figure 14: 3-node-3-layer graphlet insertion pairwise AUPRs

J Additional network distance measures

J.1 NetEmd

NetEmd [4] is a distance measure which is based on calculating earth mover's distance (EMD) between network feature distributions. Here, we use the graphlet degree distributions [2] of the networks as the features. For a set $T = \{t_1, t_2, ..., t_m\}$ of network measures, NetEmd between networks G and G' is defined as follows:

$$NetEmd_T(G, G') = \frac{1}{m} \sum_{j=1}^{m} NetEmd_{t_j}(G, G')$$

where

$$NetEmd_t(G, G') = EMD^*(p_t(G), p_t(G'))$$

where $p_t(G)$ and $p_t(G'))$ are the distributions of feature t on G and G', respectively, and

$$EMD^*(p, q) = \inf_{c \in \mathbb{R}} (EMD(\tilde{p}(\cdot + c), \tilde{q}(\cdot)))$$

where $\tilde{p}$ and $\tilde{q}$ are obtained by rescaling p and q to have variance 1, and

$$EMD(p, q) = \int_{-\infty}^{\infty} |F(x) - G(x)|$$

where $F(x)$ and $G(x)$ are the cumulative distribution functions of p and q, respectively.

Here, $p_t(G)$ is the tth graphlet degree distribution of G. Despite the name, the distribution corresponds to the tth automorphism orbit distribution within the graphlet set considered.

Here, NetEmd is implemented using pyemd [5, 6] for Python for EMD calculation and *scipy.optimize.minimize_scalar* [7] for minimization in $EMD^*(p, q)$ calculation. Pyemd requires the construction of a histogram for each of the distributions to be compared, and as such, a choice of the number of histogram bins has to be made. The number 10 for the number of bins is chosen, since it is the default for *numpy.histogram*.

The results using NetEmd (Figures 15–26) are in line with the results using graphlet correlation distance (GCD). Using multilayer graphlet degree distributions yields better precision-recall curves than using single-layer graphlet degree distributions of aggregated networks.

In Figures 16, 17, 19, 20, 22, 23, 25, and 26 we use the following convention for subfigures: **top left:** NetEmd-1-3; **top middle:** NetEmd-2-3; **top right:** NetEmd-3-3; **bottom left:** NetEmd-1-4; **bottom middle:** NetEmd-2-4; **bottom right:** NetEmd-DPK.

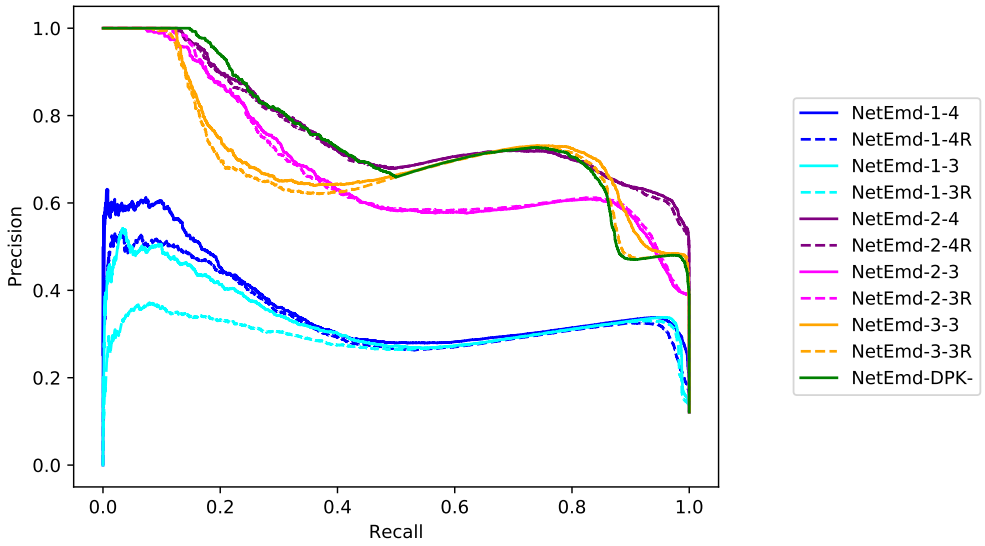


Figure 15: NetEmd Constant degree precision-recall

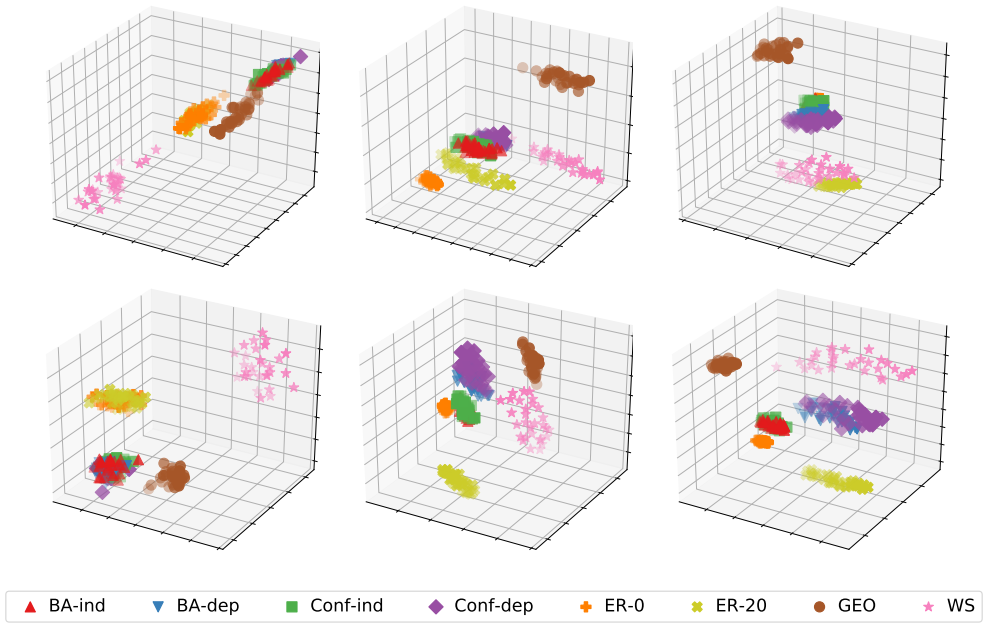


Figure 16: NetEmd Constant degree MDS

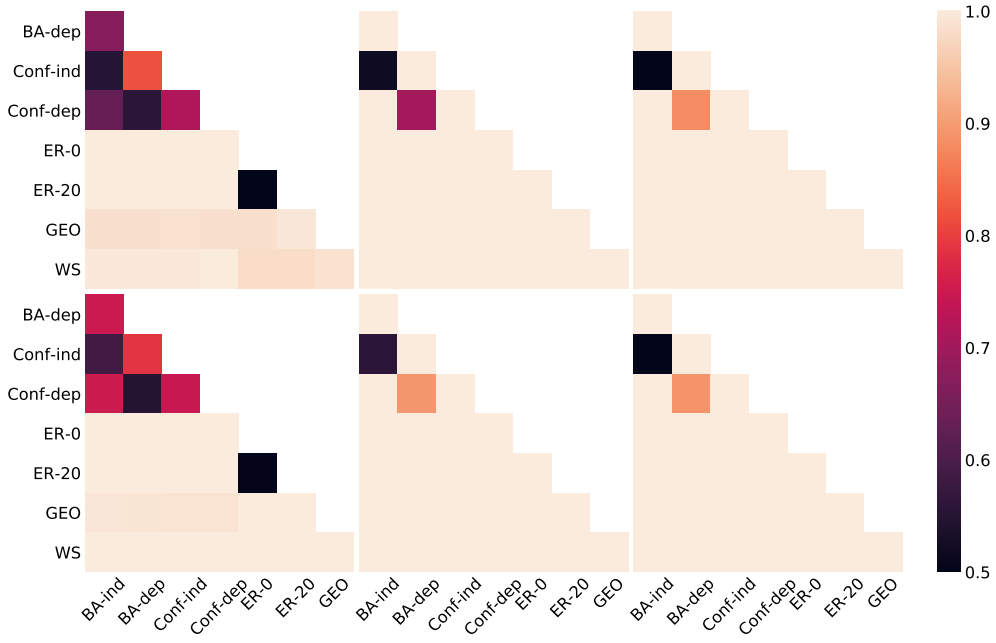


Figure 17: NetEmd Constant degree pairwise AUPRs

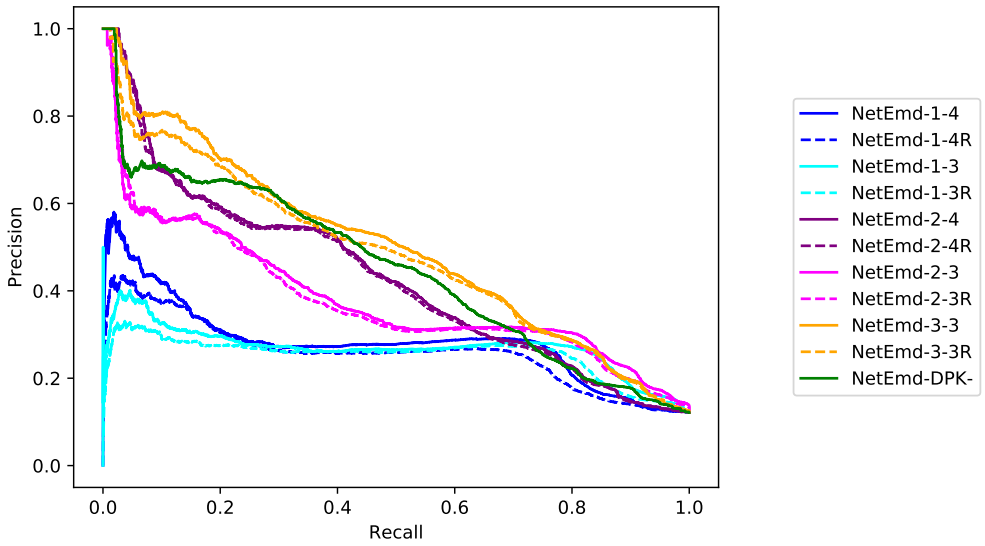


Figure 18: NetEmd Degree progression precision-recall

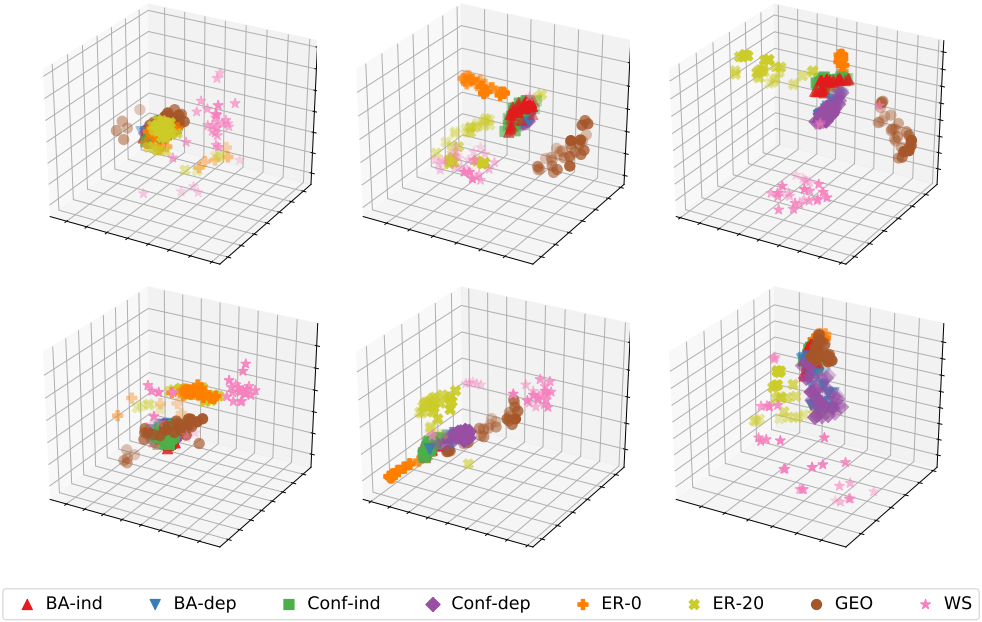


Figure 19: NetEmd Degree progression MDS

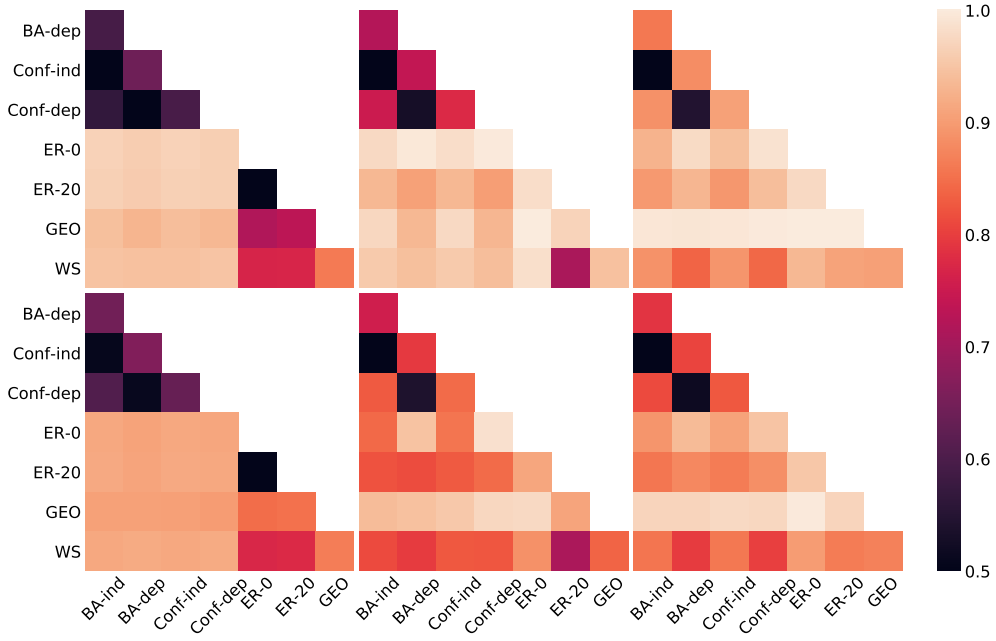


Figure 20: NetEmd Degree progression pairwise AUPRs

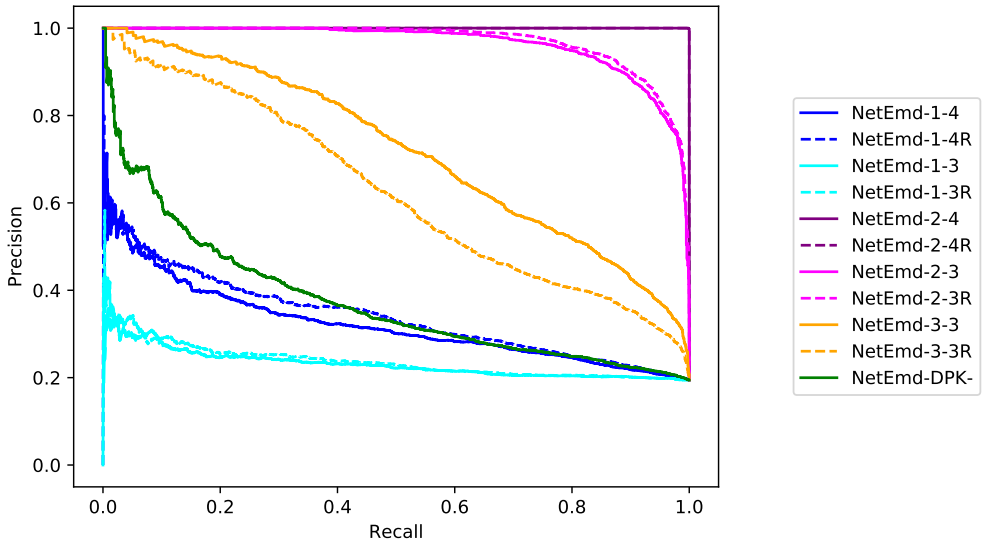


Figure 21: NetEmd 4-node-2-layer graphlet insertion precision-recall

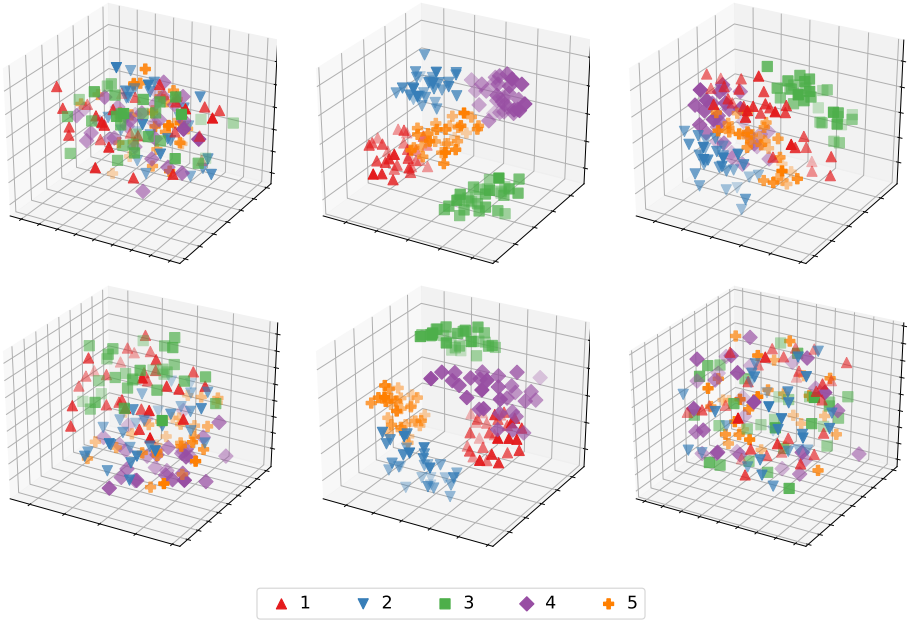


Figure 22: NetEmd 4-node-2-layer graphlet insertion MDS

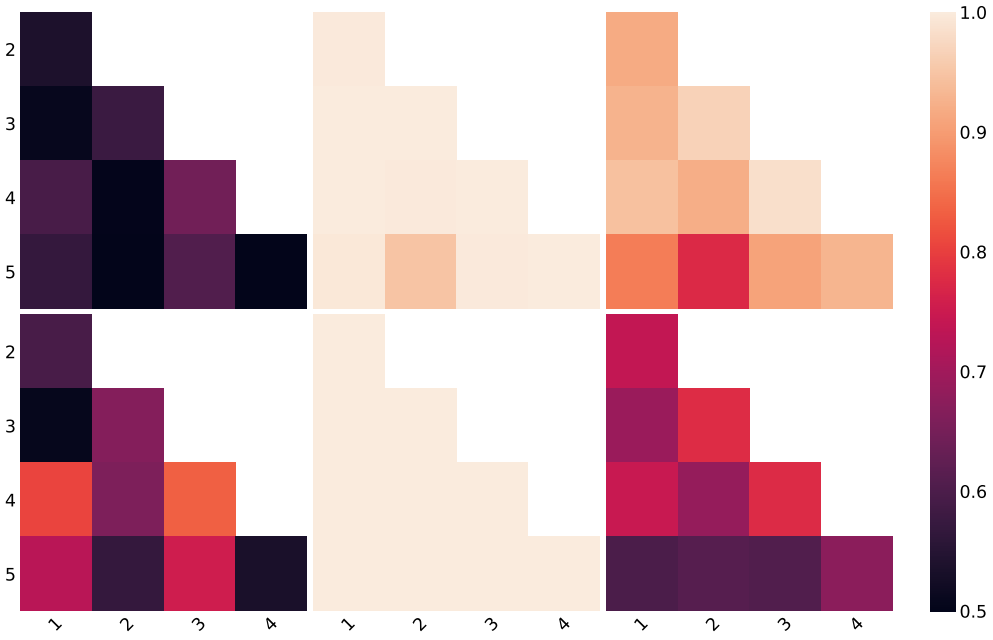


Figure 23: NetEmd 4-node-2-layer graphlet insertion pairwise AUPRs

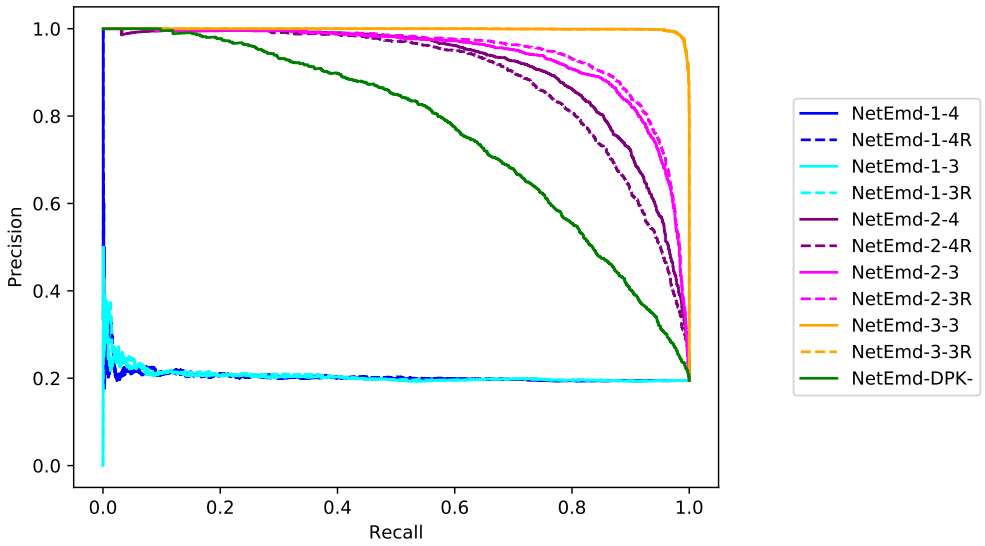


Figure 24: NetEmd 3-node-3-layer graphlet insertion precision-recall

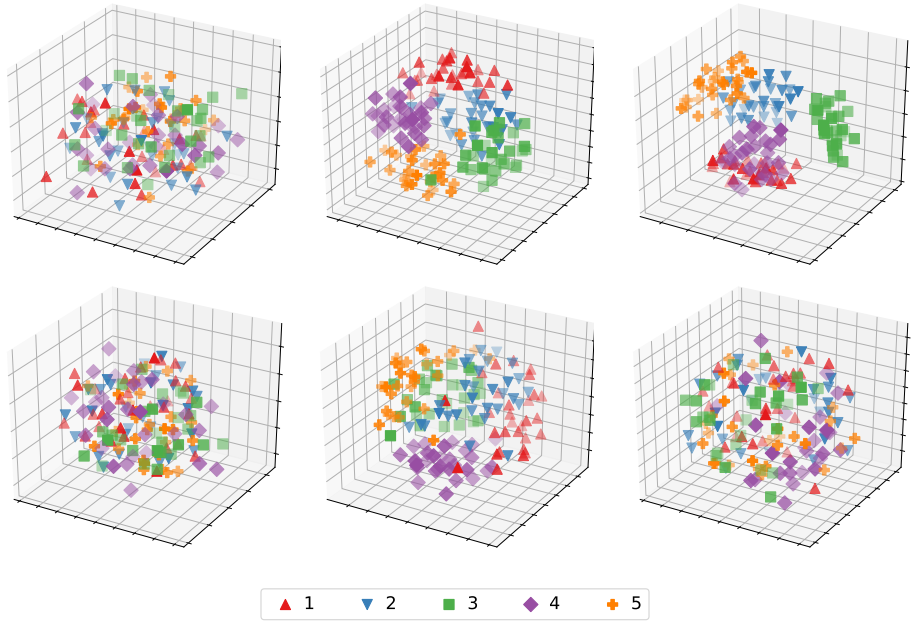


Figure 25: NetEmd 3-node-3-layer graphlet insertion MDS

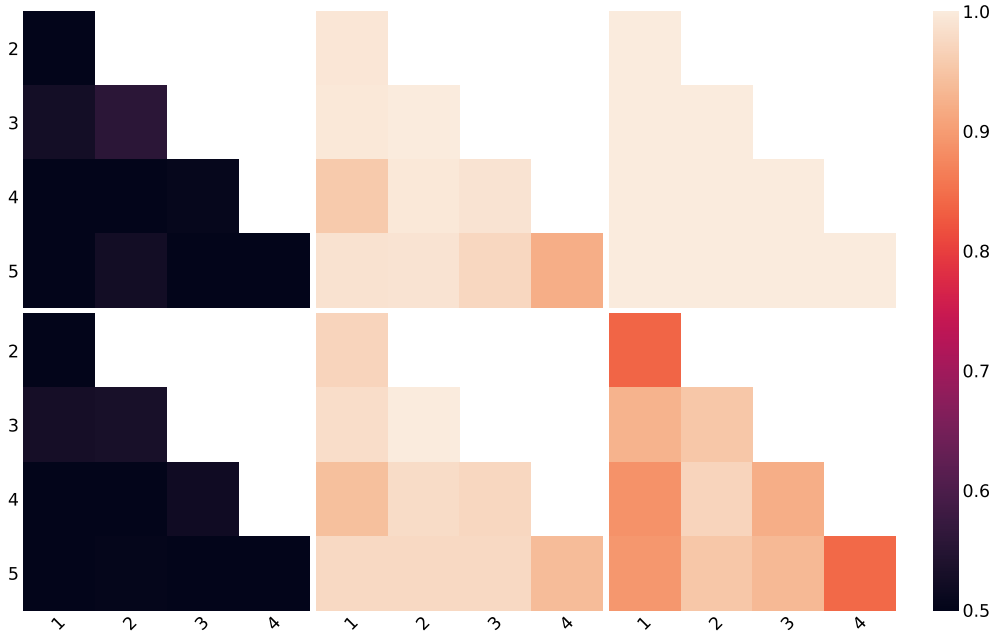


Figure 26: NetEmd 3-node-3-layer graphlet insertion pairwise AUPRs

J.2 Graphlet degree distribution agreement

Graphlet degree distribution agreement [2] (GDDA) is a distance measure which is based on calculating distances between graphlet degree distributions. It is defined as follows:

Let $d_G^t(k)$ be the tth graphlet degree distribution of network G, *i.e.* it is the distribution of the number of nodes in G on the tth automorphism orbit for a total of k times. Then,

$$S_G^t(k) = \frac{d_G^t(k)}{k}$$
$$T_G^t = \sum_{k=1}^{\infty} S_G^t(k)$$
$$N_G^t(k) = \frac{S_G^t(k)}{T_G^t}$$

and the distance between the tth orbits/graphlet degree distributions (GDDs) of networks G and G' is defined as:

$$D^t(G, G') = \frac{1}{\sqrt{2}} \left(\sum_{k=1}^{\infty} \left[N_G^t(k) - N_{G'}^t(k) \right]^2 \right)^{\frac{1}{2}}$$

and the tth GDD agreement is defined as:

$$A^t(G, G') = 1 - D^t(G, G')$$

and the agreement between networks is then defined as an average of the graphlet degree distribution agreements. Here, the arithmetic mean of the agreements is used, so that the

agreement (GDDA) between G and G' is:

$$A(G, G') = \frac{1}{m} \sum_{t=1}^{m} A^t(G, G')$$

if we index the graphlet degree distributions from 0 to $m-1$ instead of 1 to m, this becomes:

$$A(G, G') = \frac{1}{m} \sum_{t=0}^{m-1} A^t(G, G')$$

Since $A(G, G')$ is an agreement value, it is larger for networks that are more similar than for networks that are more dissimilar (and $A(G, G) = 1$). Because the other measures, GCD and NetEmd, are distances which are smaller for networks that are more similar than for networks that are more dissimilar (and $GCD(G, G) = NetEmd(G, G) = 0$), we define the GDDA distance $Adist(G, G')$ and use it in our analysis for consistency:

$$Adist(G, G') = 1 - A(G, G')$$

This is equal to taking the average of the distances $D^t(G, G')$:

$$\begin{aligned} Adist(G, G') &= 1 - A(G, G') \\ &= 1 - \frac{1}{m} \sum_{t=1}^{m} A^t(G, G') \\ &= 1 - \frac{1}{m} \sum_{t=1}^{m} \left(1 - D^t(G, G')\right) \\ &= 1 - \frac{1}{m} m + \frac{1}{m} \sum_{t=1}^{m} D^t(G, G') \\ &= \frac{1}{m} \sum_{t=1}^{m} D^t(G, G') \end{aligned}$$

The results using GDDA (Figures 27–38) are in line with the results using graphlet correlation distance (GCD). Using multilayer graphlet degree distributions yields better precision-recall curves than using single-layer graphlet degree distributions of aggregated networks.

In Figures 28, 29, 31, 32, 34, 35, 37, and 38 we use the following convention for subfigures: **top left:** GDDA-1-3; **top middle:** GDDA-2-3; **top right:** GDDA-3-3; **bottom left:** GDDA-1-4; **bottom middle:** GDDA-2-4; **bottom right:** GDDA-DPK.

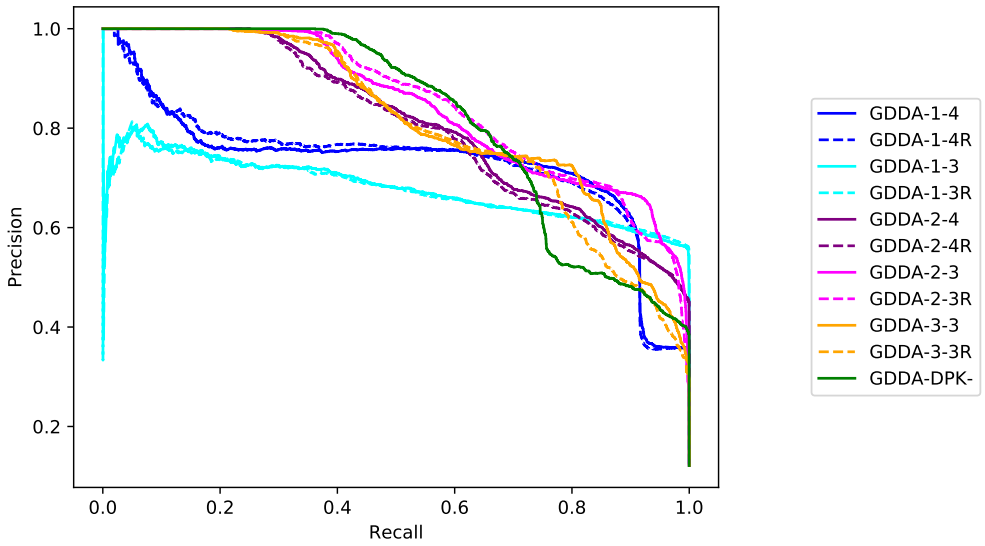


Figure 27: GDDA Constant degree precision-recall

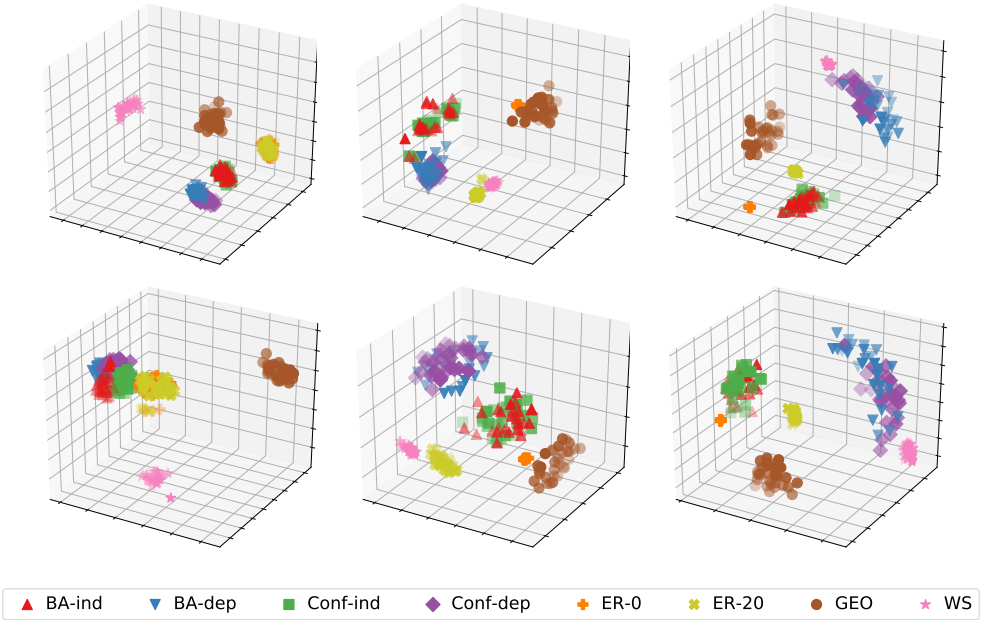


Figure 28: GDDA Constant degree MDS

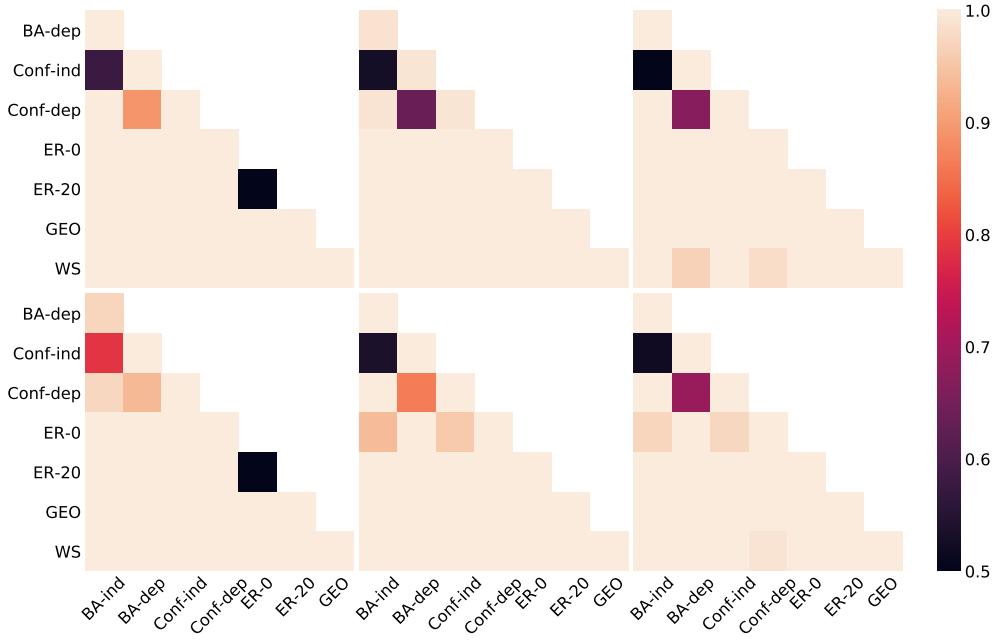


Figure 29: GDDA Constant degree pairwise AUPRs

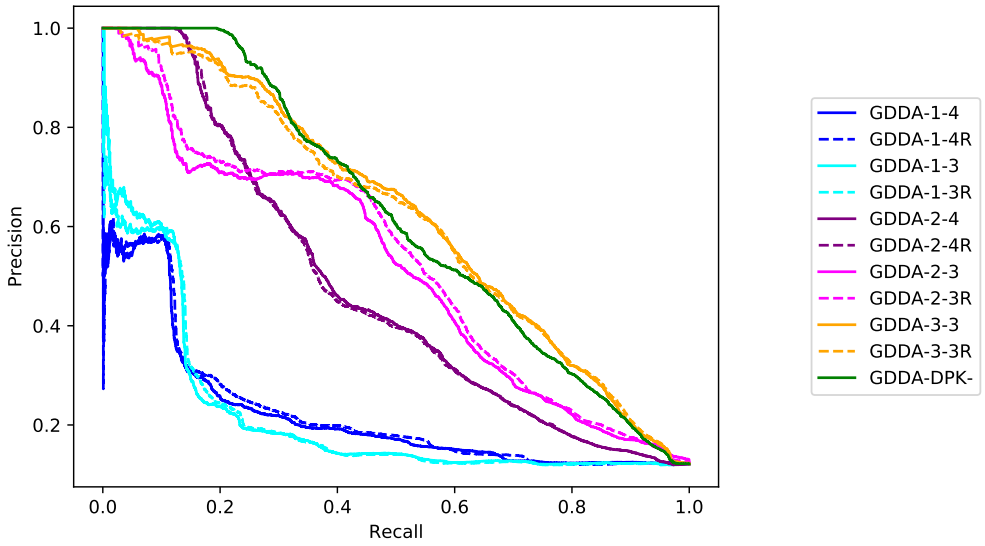


Figure 30: GDDA Degree progression precision-recall

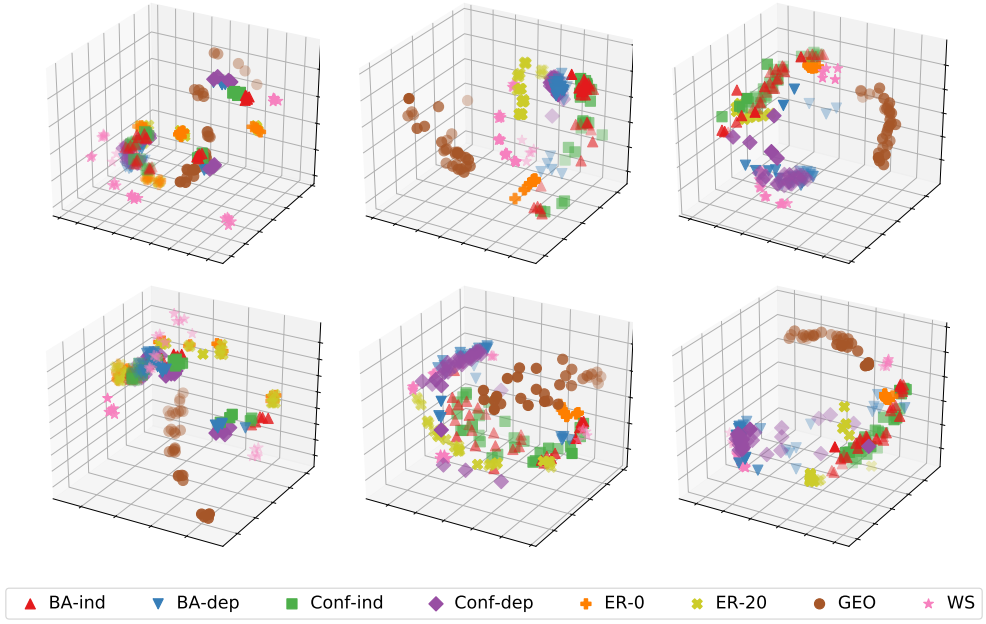


Figure 31: GDDA Degree progression MDS

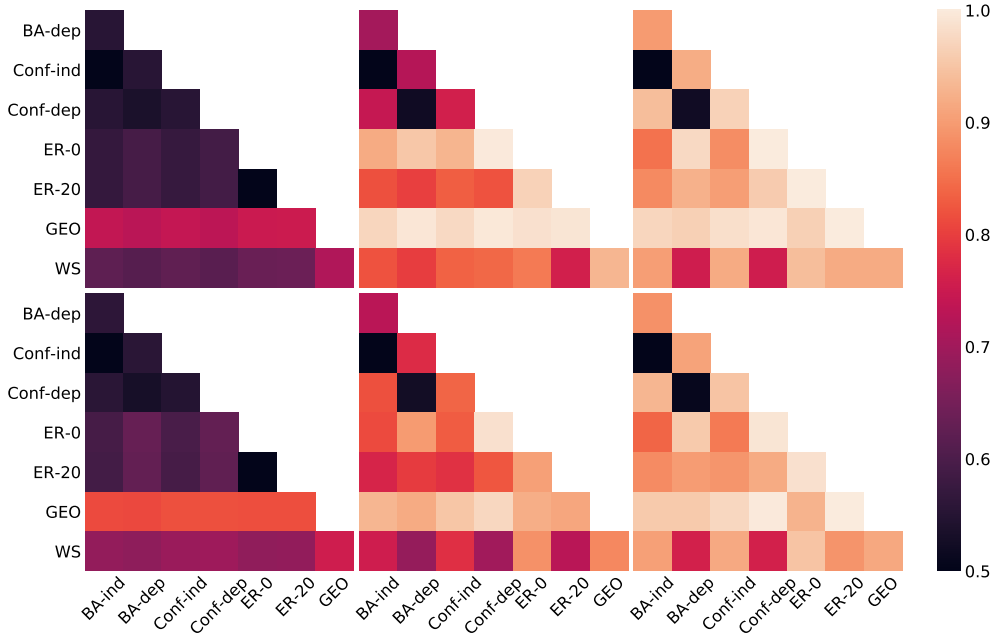


Figure 32: GDDA Degree progression pairwise AUPRs

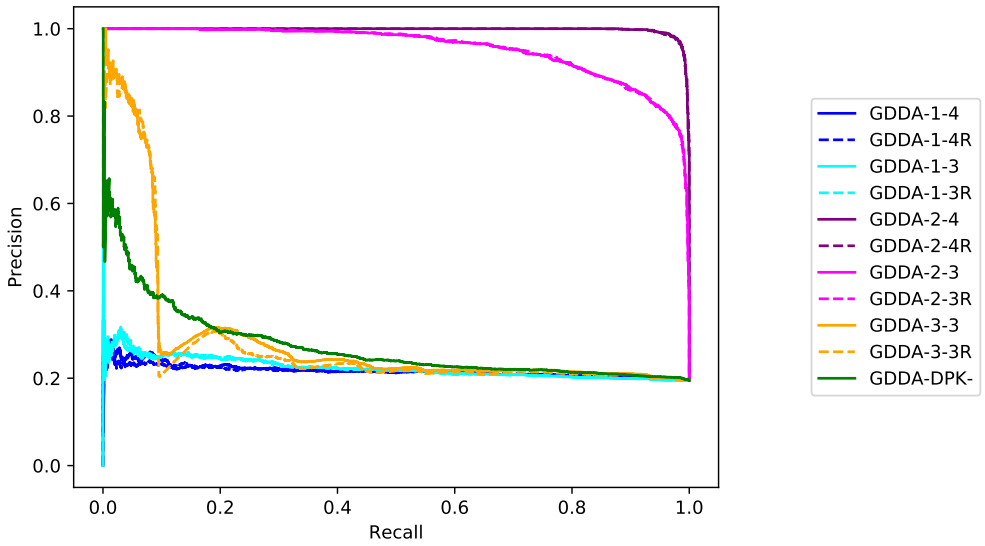


Figure 33: GDDA 4-node-2-layer graphlet insertion precision-recall

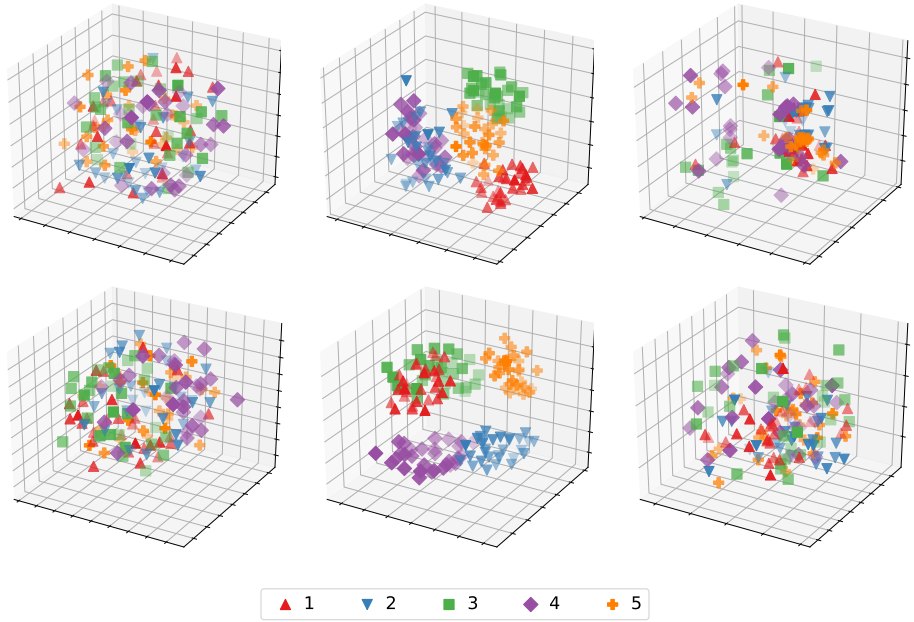


Figure 34: GDDA 4-node-2-layer graphlet insertion MDS

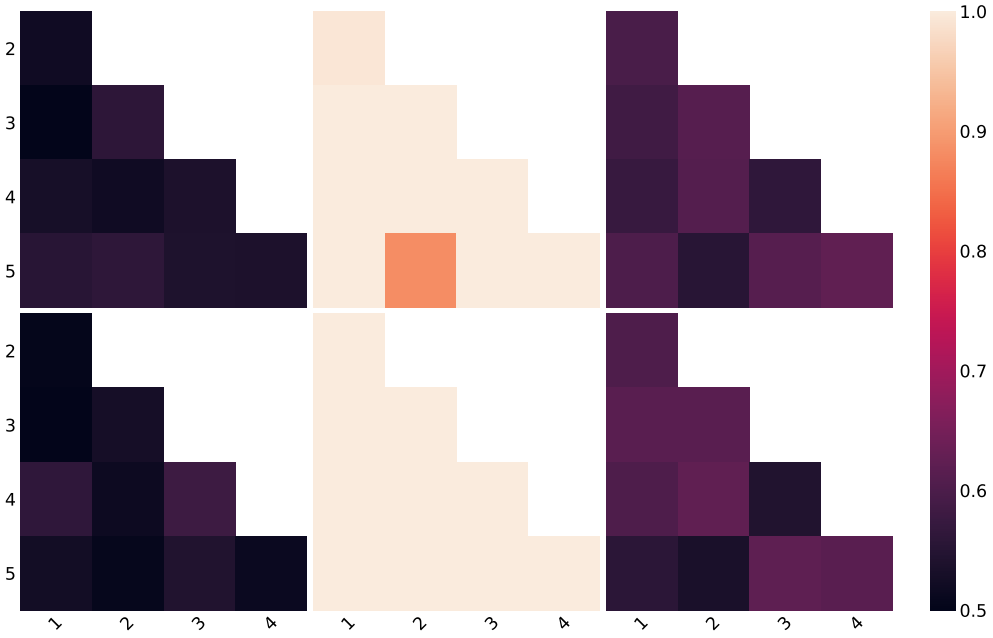


Figure 35: GDDA 4-node-2-layer graphlet insertion pairwise AUPRs

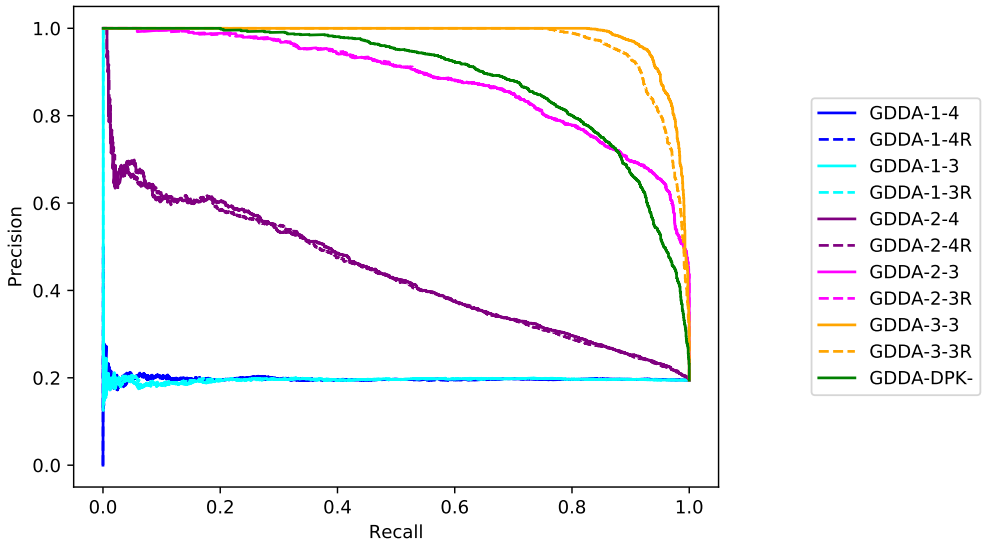


Figure 36: GDDA 3-node-3-layer graphlet insertion precision-recall

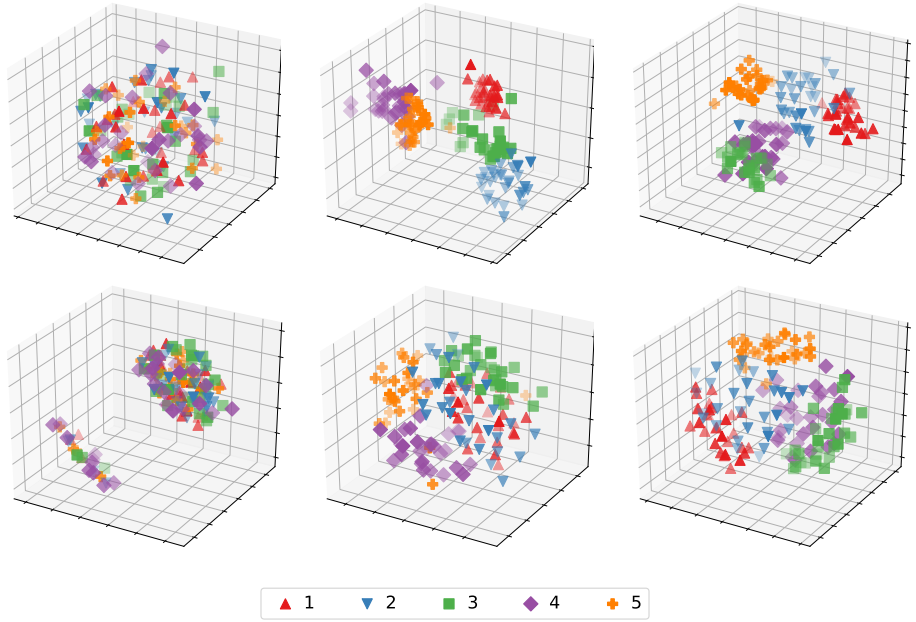


Figure 37: GDDA 3-node-3-layer graphlet insertion MDS

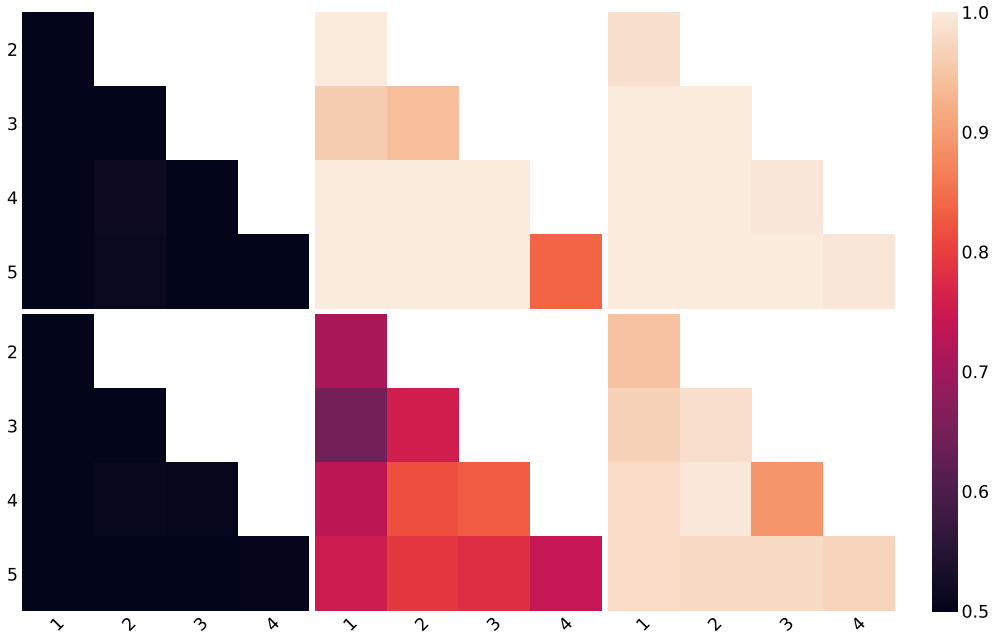


Figure 38: GDDA 3-node-3-layer graphlet insertion pairwise AUPRs

K Computational considerations

A noteworthy remark is that the running times required to compute the orbit counts increases as the number of nodes in the graphlets is increased. Another matter to consider is the number of layers to be included in the graphlets. As the number of layers in the studied networks increases, the number of layer combinations for which one needs to compute the orbit counts increases. If the number of layers in the graphlets is smaller than in the analyzed network, the summed graphlet degree vectors from different layer combinations can become difficult interpret especially if the layers depict very diverse relationships between the nodes. However, the advantage of summing the vectors from different layer combinations is that one is able to compare networks with different numbers of layers. Clearly, this summation only makes sense in a context where one uses isomorphism in which layer labels are allowed to be permuted.

L GCD performance and orbit occupancy statistics for multiplex test sets

L.1 Nonzero orbits

In order to further illustrate the reasons behind the performance difference of GCDs with different numbers of nodes and layers (see section 3.6 in the main article), we show how many orbits are occupied on average in the networks in our different network test sets, and what fraction out of all possible orbits are occupied on average (Table 2). A *nonzero orbit* is defined as an orbit where there is at least one orbit count greater than zero in a network, i.e. for node orbits there is at least one node which is on that orbit in the network. For orbits of graphlets with one layer, almost all orbits are nonzero in all test sets (average fraction of nonzeros is between 0.94 and 1.00). This is explained by the small number of orbits: when there are only few orbits, it's likely that there's at least one node on each orbit. For orbits of graphlets with two layers and three nodes, the fraction of nonzero orbits is still high (0.86-1.00) while there are many more orbits than in the one-layer-three nodes case. However, for orbits of graphlets with two layers and four nodes and three layers and three nodes, the fractions of nonzero orbits are significantly smaller (0.48-0.69), meaning that there is a large amount of orbits that have their orbit counts zero for all nodes. This could explain why GCD-2-4(R) and GCD-3-3(R) perform worse for constant intralayer degree networks than GCD-2-3(R) (see section 3.6.1 in the main article). With GCD-2-3(R), there are many orbits, providing enough information to separate the multiplex network models, while simultaneously there are few enough orbits that the fraction of nonzero orbits is high, which makes GCD calculation coherent. With GCD-1-3(R) and GCD-1-4(R) the fraction of nonzero orbits is high, but there are too few orbits to contain enough information to achieve good separation of network models; with GCD-2-4(R) and GCD-3-3(R), there are so many orbits that a significant portion of them have all orbit counts zero, containing no information and interfering antagonistically with the GCD calculation.

Table 2: Average number of nonzero node orbits (= orbits where at least one node in the network is on that orbit) and average fraction of nonzero node orbits out of all orbits for the networks in the different test sets, using node-layer isomorphism. Layers-nodes is the numbers of layers and the maximum number of nodes in the graphlets for which the orbits are evaluated. An 'R' in the end of layers-nodes indicates that redundant orbits have been removed.

		Test set			
		Different models	Degree progression	2-4-insertion	3-3-insertion
Layers-nodes	1-3	4.00 / 1.00	4.00 / 1.00	4.00 / 1.00	4.00 / 1.00
	1-3R	3.00 / 1.00	3.00 / 1.00	3.00 / 1.00	3.00 / 1.00
	1-4	14.77 / 0.98	14.81 / 0.99	15.00 / 1.00	14.16 / 0.94
	1-4R	11.00 / 1.00	11.00 / 1.00	11.00 / 1.00	11.00 / 1.00
	2-3	18.13 / 0.86	18.20 / 0.87	21.00 / 1.00	20.52 / 0.98
	2-3R	16.02 / 0.89	16.10 / 0.89	18.00 / 1.00	17.92 / 1.00
	2-4	247.13 / 0.60	276.22 / 0.67	262.92 / 0.64	195.73 / 0.48
	2-4R	231.93 / 0.62	257.04 / 0.69	252.86 / 0.68	192.15 / 0.52
	3-3	42.00 / 0.60	43.47 / 0.62	36.27 / 0.52	44.59 / 0.64
	3-3R	39.48 / 0.62	40.88 / 0.64	36.16 / 0.56	42.43 / 0.66

L.2 Orbit counts

The simple fraction of nonzero orbits doesn't explain why GCD-3-3 performs the best for the test set with progressively increasing intralayer degrees (see section 3.6.1 in the main article). The task is much harder than with constant intralayer degrees, and the performance drops for every GCD; with GCD-3-3(R), the drop is not as large as for the other GCDs, making it the best performing one. The mean orbit counts (i.e. the number for how many times a node is found on an orbit averaged over all nodes and orbits for a network), averaged over all networks in the test set, are shown in Table 3. Because the average intralayer degree is higher for the progressively increasing intralayer degree test set than the constant intralayer degree test set, the average mean orbit counts are also higher. This could have a disproportionate effect on GCD-3-3(R) over other GCDs. The average mean counts are bumped from around 3-4 to around 11-13; with the smaller count, it might be that there is not enough resolution to properly rank the orbits as is done in GCD, and with the increase, the orbit rankings might be less affected by noise and large numbers of zero orbit counts, and consequently produce better results relative to the other measures. The effect is not observed with GCD-2-4(R) which could be due to 1) the orbit counts were already high enough in the constant intralayer degree case, producing no previously unseen relative advantage with increasing intralayer degrees, and 2) there is simply more information irrelevant to the separation task in the 2-layer-4-node case than in the 3-layer-3-node case. In other words, it could be that there is relevant information in the 3-layer-3-node orbits for separating the models, but it is not manifested in the performance in the constant intralayer average test set because of the small average mean orbit count. Consequently, the increase in orbit counts might be the boosting factor that improves the relative performance of GCD-3-3 by making the relevant information contained in those orbits more noise-resistant, even with significant amounts of zero orbit counts. However, this relationship is not clear and the exact cause for GCD-3-3(R) performing relatively better in the presence of varying intralayer degrees is not

definitely resolved.

Table 3: Average mean node orbit counts for the networks in the different test sets, using node-layer isomorphism. Layers-nodes is the numbers of layers and the maximum number of nodes in the graphlets for which the orbits are evaluated. An 'R' in the end of layers-nodes indicates that redundant orbits have been removed.

		Test set			
		Different models	Degree progression	2-4-insertion	3-3-insertion
Layers-nodes	1-3	62.41	204.09	54.42	55.09
	1-3R	82.21	266.16	72.16	73.16
	1-4	611.74	3635.04	296.91	301.89
	1-4R	832.92	4939.18	404.72	411.57
	2-3	18.57	61.02	14.54	14.56
	2-3R	21.53	70.63	16.96	16.99
	2-4	25.74	155.48	10.29	10.28
	2-4R	28.41	171.58	11.36	11.35
	3-3	3.57	11.66	3.11	3.15
	3-3R	3.89	12.71	3.40	3.44

L.3 Choosing the number of nodes and layers

The choice of the number of nodes and layers in the graphlets is not obvious. On one hand, it appears that large amounts of zero orbit counts is harmful for GCD performance (Table 2), so one guiding factor could be to check the fraction of nonzero orbits and choose the highest number of nodes and layers for which the fraction is still above some threshold (e.g. ≥ 0.8). However, if there is large variance in intralayer degrees in the networks, the task becomes harder and the logic of thresholding the fraction of nonzero orbits doesn't apply. Similarly, if there are specific graphlets inserted in the networks, the best performance is achieved when using graphlets with matching number of nodes and layers for the orbits. In these cases, it seems that performance is specific to the exact graphlet content of the networks, and general guidelines for choosing the number of nodes and layers are not clear. The thresholding of the fraction of nonzero orbits could be thus used as a starting point for choosing the proper number of nodes and layers, but investigating the expected graphlet content of the specific networks in question and ensuring sufficient graphlet counts (Table 3) should be taken heavily into account.

M Application case study for real-world protein-protein interaction networks

To demonstrate the applicability of the multilayer orbit definition to analysis of real-world networks, we construct an experiment where we calculate the mean distance of real-world multiplex protein-protein interaction networks to our multiplex network models (see section 3.4 in the main article), inspired by the fitting of network models by [3]. We use 11 multiplex protein-protein interaction networks from [8] (*Arabidopsis thaliana, Bos taurus,*

Candida albicans, Caenorhabditis elegans, Drosophila melanogaster, Gallus gallus, Mus musculus, Plasmodium falciparum, Rattus norvegicus, Saccharomyces cerevisiae, Saccharomyces pombe). For the layers, we consider three interaction types that are available for all organisms, "direct interaction", "physical association", and "association", each interaction type corresponding to a layer. The mean intralayer degree for the networks is 2.089; therefore, for comparison, we use the model networks with mean intralayer degree 2. The average distances for GCD-1-3(R), GCD-2-3(R), and GCD-3-3(R) are shown in Table 4 (we limit the investigation to three-node graphlets for computational reasons). For each distance measure, the model with the smallest average distance is the best match for the data according to that measure.

According to GCD-1-3, the best match is GEO (with Conf-ind and Conf-dep distances being very close), according to GCD-1-3R, the best match is BA-dep, and according to GCD-2-3, GCD-2-3R, GCD-3-3, and GCD-3-3R, the best match is Conf-dep with a clear margin. That is, the results are consistent for the multilayer measures, and inconsistently different for the single-layer measures. This consistency indicates that the matching done with the multilayer measures is not simply by random chance, but rather that the measures are able to find relevant multilayer structure that the single-layer measures are not able to distinguish. Consequently, when modelling real-world data with multiplex/multilayer networks, it can be important to consider multilayer distance measures (such as the multilayer GCDs here) rather than aggregated single-layer measures when calculating goodness of fit.

Table 4: Average distances of multiplex network models to the multiplex protein-protein interaction networks of various organisms according to different single- and multilayer GCDs, using node-layer isomorphism. The average intralayer degree of the protein-protein interaction networks is 2.089; the model networks have average intralayer degree 2 (five networks per model).

		Multiplex network model							
		BA-ind	BA-dep	Conf-ind	Conf-dep	ER-0	ER-20	GEO	WS
GCD	1-3	0.76	0.72	0.69	0.69	1.08	1.09	**0.68**	1.19
	1-3R	0.44	**0.42**	0.50	0.46	0.76	0.77	0.51	0.71
	2-3	5.75	4.75	4.93	**3.58**	6.83	4.98	5.10	6.38
	2-3R	4.61	3.83	3.86	**3.04**	5.52	4.33	4.33	5.60
	3-3	23.47	20.15	22.22	**17.80**	26.31	21.19	29.44	23.98
	3-3R	21.54	18.07	20.35	**15.86**	24.31	19.08	27.11	21.90

Funding

This work was supported by the Academy of Finland project ECANET [320781 to MK].

References

[1] T. Dimitrova, K. Petrovski, and L. Kocarev, "Graphlets in multiplex networks," *Scientific Reports*, vol. 10, no. 1, pp. 1–13, 2020.

[2] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, 2007.

[3] Ö. N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, and N. Pržulj, "Revealing the hidden language of complex networks," *Scientific reports*, vol. 4, 2014.

[4] A. E. Wegner, L. Ospina-Forero, R. E. Gaunt, C. M. Deane, and G. Reinert, "Identifying networks with common organizational principles," *arXiv preprint arXiv:1704.00387*, 2017.

[5] O. Pele and M. Werman, "A linear time histogram metric for improved sift matching," in *Computer Vision–ECCV 2008*, pp. 495–508, Springer, October 2008.

[6] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 460–467, IEEE, September 2009.

[7] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online; accessed 2018-04-19].

[8] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature communications*, vol. 6, no. 1, pp. 1–9, 2015.