Journal of Machine Learning Research 25 (2024) 1-54 Submitted 2/23; Revised 7/24; Published 9/24

A tensor factorization model of multilayer network interdependence

Izabel Aguiar IZABEL.P.AGUIAR@GMAIL.COM
Institute for Computational and Mathematical Engineering
Stanford University
Stanford, CA 94305, USA

Dane Taylor DANE.TAYLOR@UWYO.EDU
School of Computing
Department of Mathematics and Statistics
University of Wyoming
Laramie, WY 82071, USA

Johan Ugander JUGANDER@STANFORD.EDU
Department of Management Science and Engineering
Institute for Computational and Mathematical Engineering
Stanford University
Stanford, CA 94305, USA

Editor: Qiaozhu Mei

Abstract

Multilayer networks describe the rich ways in which nodes are related by accounting for different relationships in separate layers. These multiple relationships are naturally represented by an adjacency tensor. In this work we study the use of the nonnegative Tucker decomposition (NNTuck) of such tensors under a KL loss as an expressive factor model that naturally generalizes existing stochastic block models of multilayer networks. Quantifying interdependencies between layers can identify redundancies in the structure of a network, indicate relationships between disparate layers, and potentially inform survey instruments for collecting social network data. We propose definitions of layer independence, dependence, and redundancy based on likelihood ratio tests between nested nonnegative Tucker decompositions. Using both synthetic and real-world data, we evaluate the use and interpretation of the NNTuck as a model of multilayer networks. Algorithmically, we show that using expectation maximization (EM) to maximize the log-likelihood under the NNTuck is step-by-step equivalent to tensorial multiplicative updates for the NNTuck under a KL loss, extending a previously known equivalence from nonnegative matrices to nonnegative tensors.

Keywords: multilayer networks, social networks, stochastic blockmodels, Tucker decomposition, link prediction

1. Introduction

Multilayer networks capture the many ways that a set of units can be connected: through different types of relationships in a social network (Banerjee et al., 2013; Power, 2017; Breiger et al., 1975; Sampson, 1969); at different time steps (Carlen et al., 2022; Finn et al.,

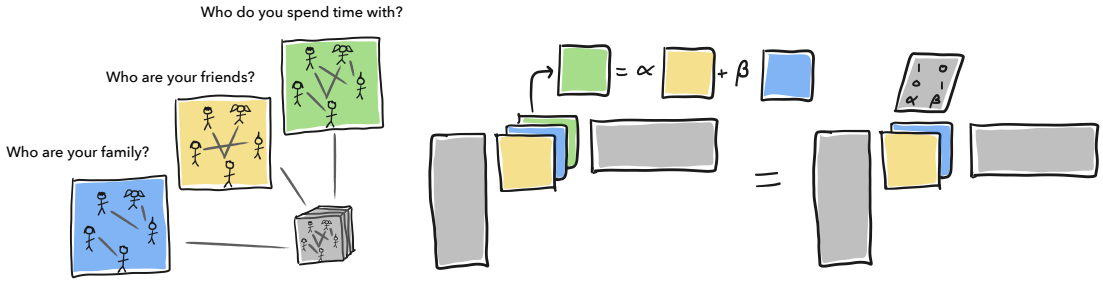


Figure 1: Multilayer networks account for the reality and variety of ways in which nodes interact in a system. In this example, a social network is complexly defined by three different types of social interaction and is represented by a tensor with three frontal slices. In this example, the process generating the "spend time with" layer is a linear combination of those processes generating the "family" and "friends" layers. On the right, we see a visual representation of the nonnegative Tucker decomposition of this network and how the third factor matrix accounts for these linearly dependent layers.

2019; Taylor et al., 2021); through different types of interactions between genes or proteins (De Domenico et al., 2015; Larremore et al., 2013); or by different modes of transit in a transportation network (De Domenico et al., 2014; Gallotti and Barthelemy, 2015). For more examples see Kivelä et al. (2014) or Boccaletti et al. (2014). As more and more data take on a multilayer network form, tools for network analysis are being steadily adapted to multilayer contexts.

Recent work has productively cast the study of multilayer community structure in the language of multilinear algebra (Wu et al., 2019), furnishing tensor-based definitions of multilayer stochastic block models (SBMs) (Schein et al., 2016; De Bacco et al., 2017; Gauvin et al., 2014; Carlen et al., 2022; Tarrés-Deulofeu et al., 2019). We extend and generalize these efforts, connecting the tensorial nonnegative Tucker decomposition (NNTuck) with KL-divergence to the statistical inference of multilayer SBMs. We show that minimizing the KL-divergence of the NNTuck is exactly equivalent to maximizing the log-likelihood of observing a multilayer network assumed to have been generated from a Poisson model with parameters defined by the NNTuck. In this sense the NNTuck identifies a natural generalization of existing multilayer SBMs, and as such, can be used for community detection and link prediction.

We investigate the use of the nonnegative Tucker decomposition to identify and statistically define layer interdependence. The vocabulary around assessing interdependence amongst the layers of a multilayer network is scattered across the literature (Battiston et al., 2014; De Domenico et al., 2015; Stanley et al., 2016). In this work, we use the term *interdependence* colloquially, to refer to the concept of dependence between layers in an abstract way without specificity about *how* the layers are dependent. Conversely, we use and define the terms of layer dependence, independence, and redundance in specific ways, as defined either by a model or by a statistical test. These terms all specify *what type* of interdependence is present in a multilayer network.

The ability to quantify and identify interdependencies between layers has the potential to inform survey instruments for collecting social network data, identify redundancies in the structure of a network, and indicate relationships between disparate layers. Reducing a multilayer network through identifying layer interdependence is both theoretically and practically appealing; although there are likely many situations where it is specifically useful, we highlight three. First, and as noted in De Domenico et al. (2015), basic structural properties of multilayer networks—like centrality, clustering coefficient, and distance—"scale superlinearly or even exponentially with the number of layers." Second, in addition to this compelling computational motivation to identify layer redundancies, there is practical motivation as well. As discussed in De Bacco et al. (2017), understanding layer redundancies in terms of social connections can help in data *collection*, as well as analysis. For example, if layers of a social network are identified as redundant, it could justify the choice to not collect those layers for future data collection in similar settings. Third, identifying redundant layers, and aggregating those layers, can help enhance structural features of networks (Nayar et al., 2015; Taylor et al., 2016, 2017).

We build upon these motivations from previous work (Schein et al., 2016; De Domenico et al., 2015; Stanley et al., 2016; De Domenico and Biamonte, 2016; De Bacco et al., 2017; Kao and Porter, 2018) and develop the NNTuck as a natural way to identify a latent space in the dimension of the layers. Analogous to how the factor matrices in the single layer SBM identify *node communities*, the additional third factor matrix in the NNTuck identifies *layer communities* (see Figure 1 for a visualization). As such, the third factor matrix of the NNTuck allows for the adjacency tensor to be low rank in the layer dimension.

Analyzing the third factor matrix is a significant focus of our work, and we propose three methods for interpreting it to quantify layer interdependence based on its structure. Furthermore, we propose definitions of layer interdependence based on likelihood ratio tests (LRTs) between different models of the data differing in the structure of that third factor matrix. To address concerns with using the traditional LRT in latent factor models, we also implement the split-LRT from Wasserman et al. (2020), which requires no regularity conditions. We use these models and tests to classify a variety of empirical networks as layer independent, dependent, or redundant, and find layer independence in a biological multilayer network, layer dependence in a cognitive social structure, and layer redundance in a collection of multilayer social support networks.

The structure of this work is as follows. In Section 2 we discuss and define the notation of stochastic block models (SBMs), multilayer networks, and previous and related approaches to multilayer SBMs. In Section 3 we define the nonnegative Tucker decomposition (NNTuck) and its notation, discuss the connection of its definition under KL-divergence to stochastic block models, motivate estimation using the multiplicative updates algorithm from Kim and Choi (2007), and offer an interpretation of the low-dimensional third factor matrix as describing the dependence between the layers of a multilayer network. In Section 4 we discuss the use of the NNTuck to empirically validate layer interdependence and define likelihood ratio test-based definitions.

In Section 5 we use cross-validation to select the NNTuck's hyper-parameters K and C. We discuss cross-validation based on two link prediction tasks: *independent link prediction*, in which elements of the adjacency tensor are missing independently and according to an identical uniform distribution, and *tubular link prediction*, in which entire tubes of the

$\boldsymbol{A}$	The $(N \times N)$ **adjacency matrix** of a single-layer network with N nodes.
$\boldsymbol{\mathcal{A}}$	The $(N \times N \times L)$ **adjacency tensor** of a multilayer network with N nodes and L layers. The **Tucker Decomposition** of $\boldsymbol{\mathcal{A}}$ is given by $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}$.
$\boldsymbol{U}$	The $(N \times K)$ ***outgoing*** **community membership matrix** in an SBM for directed networks, where K is the number of communities generating the network. For an undirected networks $\boldsymbol{U} = \boldsymbol{V}$ (see below) and we simply call $\boldsymbol{U}$ the **community membership matrix**.
$\boldsymbol{V}$	The $(N \times K)$ ***incoming*** **community membership matrix** in an SBM for a directed network. For an undirected network, $\boldsymbol{U} = \boldsymbol{V}$.
$\boldsymbol{Y}$	The **third factor matrix** in the Tucker Decomposition, also referred to as the **layer interdependence matrix**.
$\boldsymbol{G}$	The $(K \times K)$ **affinity matrix** describing the rate at which nodes in different communities form edges with one another in an SBM.
$\boldsymbol{\mathcal{G}}$	The **core tensor** in the Tucker Decomposition.

Table 1: The notation and definitions for vocabulary that will be used throughout this paper.

adjacency tensor (see Fig. 13 for a visualization of *tube fibers* in a third-order tensor) are missing (i.i.d.). In Section 6 we use the NNTuck to analyze layer dependence in practice for: two synthetic networks; the cognitive social structure dataset from Krackhardt (1987); a biological multilayer network from Larremore et al. (2013); a social support multilayer network from Banerjee et al. (2013); and 113 other multilayer social support networks from Banerjee et al. (2013, 2019). We conclude in Section 7 by discussing our work and indicating future directions of research.

2. Background

In this section we discuss related work and define notation and vocabulary. For easy reference, the primary notation is organized in Table 1. We present stochastic block models (SBMs) in Section 2.1 and nonnegative matrix factorization (NMF) in Section 2.2. In Section 2.3 we introduce tensor vocabulary and notation used throughout the work and review the Tucker decomposition. In Section 2.4 we discuss multilayer networks and in Section 2.5 we present a brief survey of related work, summarized in Table 2.

2.1 Stochastic Block Model (SBM)

Stochastic block models (SBMs) identify latent groups of nodes and the density of connections between nodes in these groups as a descriptive and/or generative tool for analyzing networks. Introduced by White et al. (1976) and expanded by Holland et al. (1983), SBMs

decompose a network into factors that aim to uncover group structure, identify to which groups each node belongs, and describe how nodes in these groups form connections with one another. Beyond these context-specific questions, SBMs identify low-dimensional structure in a network by grouping nodes into latent communities. Extensions of the original SBM have allowed for the model to account for heterogeneous degree distributions (Karrer and Newman, 2011), nodes belonging to multiple *overlapping* communities (sometimes referred to as *mixed-membership)* (Ball et al., 2011), and Bayesian approaches (Airoldi et al., 2008). Here, we focus on generalizing the *degree-corrected, mixed-membership SBM (dc-mm-SBM)* (Ball et al., 2011) to multilayer networks. Since we use much of the same framework to build our model for multilayer networks, we begin by describing the dc-mm-SBM in depth.

For a network with adjacency matrix $\boldsymbol{A} \in \mathbb{Z}_{0+}^{N\times N}$, the dc-mm-SBM assumes that each node i has outgoing and incoming nonnegative membership row vectors of dimension K, $\mathbf{u}_i$ and $\mathbf{v}_i$ respectively, describing node i's membership to K different groups when forming outgoing and incoming edges ($\mathbf{u}_i = \mathbf{v}_i$ when the network is undirected). A nonnegative *affinity matrix* $\boldsymbol{G}$ describes the rate at which nodes in different groups form edges with one another. Given $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}_{+}^{N\times K}$ and $\boldsymbol{G} \in \mathbb{R}_{+}^{K\times K}$ the dc-mm-SBM assumes each edge is an independent realization of the Poisson distribution,

$$a_{ij} \sim \text{Poisson}(\mathbf{u}_i \mathbf{G} \mathbf{v}_j^\top), \text{ for all } i, j = 1, \ldots, N.$$

For a more detailed discussion on the common modeling choice to use the Poisson distribution (as opposed to, e.g., a Bernoulli distribution), see Zhao et al. (2012). Written this way, we see that $\mathbf{u}_i \mathbf{G} \mathbf{v}_j^T$ must be positive in order to specify a valid Poisson rate. By requiring all the elements $\mathbf{u}_i, \mathbf{G}, \mathbf{v}_i$ to be independently nonnegative, the parameters are all interpretable as membership weights and affinities. In matrix form, we then have,

$$\mathbf{A} \sim \text{Poisson}(\boldsymbol{U}\mathbf{G}\boldsymbol{V}^\top). \tag{1}$$

We estimate $\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{G}$ by maximizing the log-likelihood of observing $\mathbf{A}$ under this model. Note that both weighted and unweighted networks can be described using this model and the likelihood maximized all the same.

The formulation given by (1) incorporates both the degree-corrected SBM (dc-SBM) (Karrer and Newman, 2011) and the mixed-membership SBM (mm-SBM) (Ball et al., 2011). In the dc-SBM each node may only belong to one of K groups but may have heterogeneous degree distribution. In the mm-SBM each node may belong, in part, to each of K groups but their memberships must sum to one. To account for both, the dc-mm-SBM assumes that each node has a scalar parameter $\theta_i > 0$ describing its gregariousness. Equivalently, each node's membership vector absorbs this degree parameter such that $\mathbf{u}_i = \theta_i \mathbf{s}_i$ and $\mathbf{v}_i = \theta_i \mathbf{t}_i$ for normalized membership vectors $\sum_k \mathbf{s}_k = 1$ and $\sum_k \mathbf{t}_k = 1$ for $\mathbf{s}_k, \mathbf{t}_k \geq 0$. We will henceforth describe this type of community membership as *soft membership*. Such an approach allows nodes to have membership across multiple groups while also allowing for a heterogeneous degree distribution across nodes.

There is a direct connection between the dc-mm-SBM and Poisson matrix factorization (PMF) (Gopalan et al., 2013). PMF assumes that the entries of $\boldsymbol{A}$ are realizations of a Poisson distribution with rate parameters given by the product of $\boldsymbol{W} \in \mathbb{R}_{+}^{N\times K}$ and

$\boldsymbol{H} \in \mathbb{R}_+^{K \times N}$. That is, $a_{ij} \sim \text{Poisson}\left(\sum_k w_{ik} h_{kj}\right)$. Dropping the constant term, the log-likelihood of observing $\boldsymbol{A}$ under this distribution is given by,

$$\mathcal{L}(\boldsymbol{A}|\boldsymbol{W}, \boldsymbol{H}) = \sum_{ij} a_{ij} \log \sum_k w_{ik} h_{kj} - \sum_k w_{ik} h_{kj}. \tag{2}$$

The factors $\boldsymbol{U}, \boldsymbol{G}$, and $\boldsymbol{V}$ can be grouped together such that we can consider the dc-mm-SBM as equivalent to the Poisson matrix factorization (e.g., if we define $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{G}$ and $\boldsymbol{H} = \boldsymbol{V}^\top$, then the nonnegative factors of the dc-mm-SBM also describe the nonnegative factors of a Poisson matrix factorization).

2.2 Nonnegative Matrix Factorization (NMF)

A related approach for finding latent structure in a matrix, nonnegative matrix factorization (NMF) (Paatero and Tapper, 1994; Lee and Seung, 1999), aims to factor nonnegative matrix $\boldsymbol{A}$ into two nonnegative factor matrices, $\boldsymbol{W}$ and $\boldsymbol{H}$, for $\boldsymbol{W} \in \mathbb{R}_+^{N \times K}$ and $\boldsymbol{H} \in \mathbb{R}_+^{K \times N}$.

Not every matrix can be exactly factorized in this way, and although for such cases it is more accurately described as nonnegative matrix *approximation*, we will henceforth refer to both problems as NMF. When estimating the NMF of a matrix $\boldsymbol{A}$ there are many loss functions with respect to which the factorization may be optimized. For reasons that will be clearly motivated in the following sections, we focus on minimizing the *KL-divergence* between the matrix and its factorization, defined as

$$\text{KL}(\boldsymbol{A} \| \boldsymbol{W}\boldsymbol{H}) = \sum_{ij} \left(a_{ij} \log \frac{a_{ij}}{(\boldsymbol{W}\boldsymbol{H})_{ij}} - a_{ij} + (\boldsymbol{W}\boldsymbol{H})_{ij} \right). \tag{3}$$

An algorithm based on multiplicative updates for minimizing KL-divergence was developed by Lee and Seung (2000) and is widely used to find local optima of the non-convex optimization problem given by Eq. (3). This algorithm guarantees that, given nonnegative initializations, factor matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ remain nonnegative throughout the optimization. Furthermore, the algorithm guarantees monotonic convergence to a local minimum.

It is known that *maximizing* the log-likelihood in PMF Eq. (2) is equivalent to *minimizing* the KL-divergence in NMF:

$$\begin{aligned}
&\underset{\boldsymbol{W},\boldsymbol{H}}{\text{minimize}} \ \text{KL}(\boldsymbol{A}||\boldsymbol{W}\boldsymbol{H}) \\
&\Leftrightarrow \underset{\boldsymbol{W},\boldsymbol{H}}{\text{minimize}} \ \sum_{ij} (a_{ij} \ \text{log} a_{ij} - a_{ij} \log(\boldsymbol{W}\boldsymbol{H})_{ij} - a_{ij} + (\boldsymbol{W}\boldsymbol{H})_{ij}) \\
&\Leftrightarrow \underset{\boldsymbol{W},\boldsymbol{H}}{\text{minimize}} - \sum_{ij} (a_{ij} \text{log}(\boldsymbol{W}\boldsymbol{H})_{ij} - (\boldsymbol{W}\boldsymbol{H})_{ij}) \\
&\Leftrightarrow \underset{\boldsymbol{W},\boldsymbol{H}}{\text{maximize}} \ \mathcal{L}(\boldsymbol{A}|\boldsymbol{W}, \boldsymbol{H}).
\end{aligned} \tag{4}$$

Furthermore, as was first noted in Févotte and Cemgil (2009), using expectation maximization (EM) to find a local maximum of the log-likelihood for PMF is step-by-step equivalent to using the multiplicative updates given in Lee and Seung (2000) to minimize KL-divergence. This equivalence does not hold when comparing EM updates under a Gaussian generative

model to the multiplicative updates under a Frobenius loss. This observation, in combination with the equivalence in Eq. (4), gives NMF with KL-divergence a strong statistical foundation. Specifically, there are two important connections to be made: (i) to *factorize* matrix $\boldsymbol{A}$ into a product of two nonnegative matrices by maximizing log-likelihood in PMF and minimizing KL-divergence in NMF are equivalent optimization problems, and (ii) the *algorithms* (Lee and Seung's multiplicative updates and EM) by which to find the model associated with a local minimum of the shared loss function are the same for PMF and NMF.

2.3 Tensor Notation and Tucker Decomposition

To facilitate a clear analysis and discussion of the tensor-based model in Section 3, we now define the tensor-specific vocabulary and notation that we will use throughout this paper. For a more thorough overview of tensor vocabulary, methods, decompositions, and definitions, see Kolda and Bader (2009) for an excellent review. We focus notation and terms to *third-order* "frontally square" tensors $\boldsymbol{\mathcal{X}}$ of dimension $N \times N \times L$.

Frontal slices The frontal slices of $\boldsymbol{\mathcal{X}}$ are the L matrices of size $N \times N$ that, when stacked together, form the $N \times N \times L$ tensor. A depiction of frontal slices can be found at the bottom left of Fig. 12. We denote the ℓth frontal slice of $\boldsymbol{\mathcal{X}}$ as $\mathbf{X}_\ell$. The frontal slice of an adjacency tensor $\boldsymbol{\mathcal{A}}$ corresponds to the adjacency matrix $\boldsymbol{A}_\ell$ of a particular layer ℓ of the multilayer network, and thus we will make frequent references to it.

Tensor fibers Analogous to rows and columns in matrices, third-order tensors have what are called *row*, *column*, and *tube* fibers, denoted $\boldsymbol{\mathcal{X}}_{:jk}$, $\boldsymbol{\mathcal{X}}_{i:k}$, and $\boldsymbol{\mathcal{X}}_{ij:}$, respectively. See Figure 13 for a visualization of each.

Unfoldings A third-order tensor has three unfoldings: the 1-unfolding, 2-unfolding, and 3-unfolding. These are higher-dimensional equivalents to *vectorizing* a matrix. The n-unfolding of a third-order tensor stacks its column, row, or tube fibers to form a matrix, and is denoted by $\mathbf{X}_{(n)}$. See Figure 12 and Section 2 of Kolda and Bader (2009) for helpful visualizations.

The tensor n-mode product ($\times_n$) A third-order tensor can be multiplied by a matrix through a 1-, 2-, or 3-mode product. Dimensionally, for an $N \times N \times L$ tensor $\boldsymbol{\mathcal{X}}$ one can take the 1-mode product with a $P \times N$ matrix, the 2-mode product with a $Q \times N$ matrix, and the 3-mode product with a $R \times L$ matrix. The resulting dimensions of these mode products would be $P \times N \times L$, $N \times Q \times L$, and $N \times N \times R$, respectively. Elementwise, the 1-mode product gives $(\boldsymbol{\mathcal{X}} \times_1 \mathbf{B})_{ijk} = \sum_h x_{hjk} b_{ih}$.

Tucker decomposition Although the most prominent of the many tensor decompositions are the CP decomposition (Carroll and Chang, 1970; Harshman, 1970) and the Tucker decomposition (Tucker, 1966), other notable decompositions include RESCAL (Nickel et al., 2011), DEDICOM (Harshman, 1978), and PARATUCK2 (Harshman and Lundy, 1996), where both the CP and RESCAL decompositions are special cases of the Tucker decomposition. The Tucker decomposition decomposes an nth-order tensor $\boldsymbol{\mathcal{X}}$ into an nth-order *core tensor* $\boldsymbol{\mathcal{G}}$ and n *factor matrices*. The Tucker decomposition of a third order $N \times N \times L$

tensor $\boldsymbol{\mathcal{X}}$ is

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}. \tag{5}$$

Here $\boldsymbol{\mathcal{G}}$ is $P \times Q \times R$, $\boldsymbol{U}$ is $N \times P$, $\boldsymbol{V}$ is $N \times Q$, and $\boldsymbol{Y}$ is $L \times R$. A *nonnegative Tucker decomposition* is one where all elements of the factor matrices $\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{Y}$ and core tensor $\boldsymbol{\mathcal{G}}$ are nonnegative. When fitting a nonnegative Tucker decomposition to a data tensor $\boldsymbol{\mathcal{X}}$, two common notions of approximation are the Frobenius loss and the KL-divergence, where the latter is given by

$$\mathrm{KL}(\boldsymbol{\mathcal{X}} \mid \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}) := \sum_{ijk} x_{ijk} \log \frac{x_{ijk}}{\hat{x}_{ijk}} - x_{ijk} + \hat{x}_{ijk}, \tag{6}$$

for $\widehat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}$.

2.4 Multilayer Networks

Multilayer networks consist of a set of different network 'layers' which each encode different types of edges (sometimes called *intralayer* edges). These types of edges can represent, for example, different types of relationships in social networks (Banerjee et al., 2013; Power, 2017; Banerjee et al., 2019), different shared genetic subsequences in biological networks (Larremore et al., 2013), or different time steps in temporal networks (Gallotti and Barthelemy, 2015). In general, the node set can differ across layers, however we focus on a subclass called *multiplex networks* in which the nodes are identical in each layer (Mucha et al., 2010).

Distinct from *heterogeneous networks* (e.g., Dong et al., 2020), wherein there are different categories of nodes *and* edges, multilayer networks have only one type of node and only distinguish between different types of edges. Similarly, *multigraphs* allow for multiple edges to exist between nodes and *labels* corresponding to nodes and/or edges identify the different types of relationships. For a more comprehensive survey of related terminologies, see Kivelä et al. (2014).

Although some work about multilayer networks also models connections between layers using *interlayer* edges, we do not assume or model the coupling of layers. In this case, there exists a one-to-one alignment of layers, allowing them to be encoded in a 3-dimensional *adjacency tensor* $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N \times N \times L}$, where N and L are the numbers of nodes and layers, respectively, and where each frontal slice $\boldsymbol{A}_\ell$ is the adjacency matrix of a particular layer ℓ of the multilayer network. Here, $a_{ij\ell} > 0$ if an only if there is an edge from i to j in layer ℓ, and is otherwise zero.

Multilayer networks can be either directed or undirected. In this work we assume that all layers within a given network are either directed or undirected. Extending our approach to networks that have a mixture of directed and undirected layers would be straightforward. We also assume that all edges are unweighted, $a_{ij\ell} \in \{0, 1\}$, but this assumption can easily be relaxed. For a more comprehensive review of multilayer networks, see Kivelä et al. (2014); Boccaletti et al. (2014)

2.5 Related Work

We now discuss related work as categorized by previous approaches for (i) tensor methods for multilayer networks, (ii) stochastic block models for multilayer networks, and (iii) addressing layer interdependence in multilayer networks.

Tensor methods for multilayer networks Multilayer networks have been studied since as far back as 1939 (Roethlisberger and Dickson, 1939), and they have been mathematically represented by tensors since at least 1987, when Krackhardt introduced the concept of *cognitive social structures* (Krackhardt, 1987). In fact, one of the foundational tensor decomposition papers by Carroll and Chang (1970), although not a multilayer network, was a study of multilayer relational data. Since then, tensor methods have become more prominent in analysing multilayer networks (Bader et al., 2007; Kolda and Bader, 2009; Nickel et al., 2011), and De Domenico et al. (2013) formalized this tensorial framework by generalizing many network analysis tools to the multilayer setting.

The CP tensor decomposition (Carroll and Chang, 1970; Harshman, 1970) and the Tucker decomposition (Tucker, 1966), have been implemented for their use in analyzing multilayer networks. The CP decomposition, for example, is implemented to interpret a fourth-order tensor of multilayer network data in Schein et al. (2015), for community detection and analysis of activity patterns in a temporal network in Gauvin et al. (2014), and to assess centrality of nodes in multilayer networks Wang and Zou (2018). The Tucker decomposition is used for community detection in a temporal multilayer network representing brain dynamics in Al-Sharoa et al. (2018) and to cluster keywords and communities in a multilayer email network in Sun et al. (2009).

Stochastic block models for multilayer networks There have been a wide range of approaches to generalize the SBM to multilayer networks. In Valles-Catala et al. (2016) a multilayer SBM is developed by fitting a new SBM to each layer, assuming that neither node-membership nor group-to-group connectivity is fixed across layers. Stanley et al. (2016) develop a related model that assumes layers are sampled from a small set of SBMs and the set of layers generated from the same SBM are referred to as belonging to the same *strata*. In Carlen et al. (2022) and De Bacco et al. (2017), a node's membership vectors are held fixed across layers, but a new affinity matrix is fit for each layer. A similar model is proposed in Paul and Chen (2016) but with node membership vectors constrained to take on binary values and with a Bernoulli distribution assumption instead of Poisson. Conversely, in Tarrés-Deulofeu et al. (2019) a Tucker decomposition accounting for layer community structure is fit with the aim to predict *types* of links in a multilayer network. To do so, a new core tensor is fit for each type of link. Although layer community structure is addressed in Tarrés-Deulofeu et al. (2019), the number of node-communities is always fixed to equal the number of layer-communities: a missed opportunity to examine layer interdependence by examining the optimal number of layer-communities.

In Wang and Zeng (2019), the authors propose using a Tucker decomposition as a multilayer SBM, but limit their factor matrices to only take on binary values. Thus, the extent to which layer dependence is addressed is limited to the binary clustering of layers and is more similar to the strata work of Stanley et al. (2016). Furthermore, the core tensor is not constrained to be nonnegative, and the proposed algorithm focuses on minimizing

the Frobenius norm of the difference between the tensor and the approximation given by the Tucker decomposition.

Previous work explicitly using the full nonnegative Tucker decomposition as a multilayer extension of the SBM to study layer dependence is limited to that of Schein et al. (2016), wherein the authors propose the use of a Bayesian Poisson Tucker Decomposition (BPTD) as a generalization of the dc-mm-SBM to study a multilayer network. They highlight the BPTD using a fourth-order tensor to study international relations between countries over time, and show how the BPTD can group together countries, actions, and time periods into communities. Our work extends this modeling framework by introducing a technical approach to studying layer dependence. We build upon Schein et al. (2016)'s work to significantly expand the motivation, estimation, and interpretation of the nonnegative Tucker decomposition as a sensible extension of the SBM to multilayer networks. Distinct from their work, we aim to motivate the nonnegative Tucker decomposition as a model for understanding layer dependence *in general*, and we do so through extensive examples (see Section 3.1). Furthermore, departing from their MCMC algorithm, we justify the use of an algorithm for estimating a nonnegative Tucker decomposition by minimizing the KL-divergence with multiplicative updates (Kim and Choi, 2007) by connecting it to the pointwise maximum likelihood estimate of the log-likelihood using expectation maximization—the estimation method proposed in De Bacco et al. (2017).

Because we will often reference the work and the multilayer SBM model MULTITENSOR (MT) built in De Bacco et al. (2017), we define and discuss the work in more detail here. Consider a multilayer network with N nodes and L layers represented by adjacency tensor $\boldsymbol{\mathcal{A}} \in \mathbb{Z}_{0+}^{N\times N\times L}$. Assume each node i in the network has outgoing and incoming nonnegative membership vectors $\mathbf{u}_i$ and $\mathbf{v}_i$, respectively, representing their soft assignment to K groups. The densities with which nodes in each community interact in layer ℓ is given by nonnegative affinity matrix $\boldsymbol{G}_\ell$. The MT model then assumes the generative process whereby

$$\boldsymbol{\mathcal{A}} \sim \text{Poisson}(\boldsymbol{\Lambda}), \text{ where } \boldsymbol{\Lambda}_\ell = \boldsymbol{U}\boldsymbol{G}_\ell\boldsymbol{V}^\top \text{ for } \ell \in [1, L]. \tag{7}$$

Written this way we see that MT fits an SBM to *each* layer of the network, holding fixed the outgoing and incoming group memberships across layers. Parameters $\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{G}_\ell$ are estimated by maximizing the log-likelihood of observing $\boldsymbol{\mathcal{A}}$ via an EM algorithm.

Layer interdependence Understanding how the layers of a multilayer network interact with, represent, or are different from one another has been a relevant question ever since multilayer networks started being studied. As such, there have been a multitude of proposed methods to study and assess layer interdependence. Krackhardt (1987) suggested differentiating layer similarity by comparing individual layers to a *consensus structure*. Battiston et al. (2014) introduce the measure of *edge overlap* which they propose to use for determining similarity between layers. This measure is built upon in Kao and Porter (2018) which uses a similarity measure based on edge overlap to identify layer communities. The authors construct a single layer network where each node is a layer and each edge is weighted by the similarity measure, and then find layer communities by doing community detection on this new network. De Domenico et al. (2015) and De Domenico and Biamonte (2016) develop information-theoretic tools to identify layer dependency and cluster similar layers. In Stanley et al. (2016), the authors study layer interdependence by categorizing layers into

groups such that all layers were drawn from the same SBM. In the MULTITENSOR (MT) work of De Bacco et al. (2017), layer interdependence is studied through building multiple MT models using different subsets of the layers, and the models' performance (measured by test-AUC) on a link prediction task is used to determine if there is layer interdependence in the model. In this setting, layer interdependence can be viewed as a specific application of transfer learning which assesses how a model built in one setting performs in an alternative one (Torrey and Shavlik, 2010; Altenburger and Ugander, 2021). Finally, while not explicitly developed for use in the multilayer setting, tools to compare similar structures across graphs such as those developed in Wills and Meyer (2020) and Racz and Sridhar (2021) could be used to compare layers in a multilayer network.

These various approaches to studying layer interdependence have been applied to various disciplinary contexts, and have resulted in varying discipline-specific conclusions, as well. In particular: De Domenico et al. (2015) identifies and interprets layer dependence in varying contexts—from the worldwide food import/export multilayer network to the London metropolitan public transportation multilayer network; Battiston et al. (2014) interprets the dependencies between trust and communication, business, and operating partnerships within an Indonesian terrorist network; Kao and Porter (2018) interprets the dependencies between different research areas in the American Physical Society's collaboration network, the regional dependencies in an airline network, and the distinction between positive relationships, negative relationships, and temporally distinguished esteem in a social network; Stanley et al. (2016) and De Domenico and Biamonte (2016) both discuss the interpretation of layer dependence in the context of the human microbiome, and the ability to use layer interdependence methods to identify similarly functioning regions within the body. Overall, previous work on studying layer interdependence in multilayer networks identifies many disciplines and contexts within which such methods have the potential to corroborate or identify socially, scientifically, or theoretically interesting findings. We aim to add to this body of work.

In contrast to these previous approaches to identify layer interdependence, we propose that the nonnegative Tucker decomposition does so by identifying which layers can be described by shared generative stochastic block models. As we will discuss in Section 3.2, the NNTuck is a generalization of the strata SBM model from Stanley et al. (2016) that is similar to moving from fixed- to mixed-membership assignments in the single layer SBM.

2.6 Contributions

Situated in this related work, the contributions of our work are as follows: (i) we use and expand the motivation of the nonnegative Tucker decomposition with KL-divergence as a natural extension of the dc-mm-SBM to multilayer networks by allowing for distinct latent structure in the nodes *and* layers; (ii) we propose the NNTuck as a generalization of many prior models and approaches for extending the SBM to multilayer networks (see Table 2 to see how the NNTuck generalizes related work); (iii) we propose the inspection of the third factor matrix in the NNTuck for quantifying and assessing layer interdependence and discuss three specific methods for doing so; (iv) we show the equivalence in model, loss function, and algorithm between Poisson Tucker decomposition and nonnegative Tucker decomposition; (v) we propose definitions of layer interdependence based on the likelihood ratio test; (vi) we

define two link prediction tasks for multilayer networks to use cross-validation as a tool for model selection; and (vii) we use the NNTuck to study layer interdependence in a variety of empirical multilayer networks: one biological, one cognitive social structure, and 113 social support networks.

3. Nonnegative Tucker Decomposition (NNTuck)

We begin this section by outlining our approach to a multilayer SBM that corresponds to a nonnegative Tucker decomposition with KL-divergence. We will henceforth refer to the multilayer SBM developed here as just the nonnegative Tucker decomposition (NNTuck), although it's important to note that the SBM interpretation only corresponds to the nonnegative Tucker decomposition estimated with KL-divergence loss as in Equation (6), *not* Frobenius loss. We present the model of the NNTuck as a multilayer SBM in Section 3.1, and discuss *deflation* and layer dependence in the NNTuck in Section 3.2. We present an algorithm for estimating the NNTuck of a multilayer network and discuss the algorithm's limitations in Section 3.3.

3.1 The Model

Consider a multilayer network with N nodes and L layers represented by adjacency tensor $\boldsymbol{\mathcal{A}} \in \mathbb{Z}_{0+}^{N\times N\times L}$. We assume that each node i has nonnegative membership vectors $\mathbf{u}_i \in \mathbb{R}_+^K$ and $\mathbf{v}_i \in \mathbb{R}_+^K$ representing its soft assignment to $K \leq N$ groups. Moreover, we assume that each layer ℓ has nonnegative vector $\mathbf{y}_\ell \in \mathbb{R}_+^C$ describing the layer's soft membership to each of $C \leq L$ layer communities. Just as matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ in the SBM describe latent community structure in the *nodes* of single-layer networks, the factor matrix $\boldsymbol{Y}$ in the NNTuck describes latent structure in the *layers* of a multilayer network. Finally, we assume tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{K\times K\times C}$ defines C different affinity matrices. Let $\mathbf{u}_i, \mathbf{v}_i, \mathbf{y}_\ell$ be the rows of nonnegative matrices $\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{Y}$, respectively. The NNTuck multilayer SBM assumes

$$\boldsymbol{\mathcal{A}} \sim \text{Poisson}(\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}). \tag{8}$$

For an undirected network, we set $\boldsymbol{U} := \boldsymbol{V}$ and constrain the frontal slices of $\boldsymbol{\mathcal{G}}$ to be symmetric. Maximizing the log-likelihood of observing $\boldsymbol{\mathcal{A}}$ under the model given by (8) is equivalent to minimizing the KL-divergence between $\boldsymbol{\mathcal{A}}$ and $\hat{\boldsymbol{\mathcal{A}}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}$. This is a tensorial generalization of the connection between PMF and NMF with KL-divergence referenced in (4) and motivates the use of the KL-divergence for determining the NNTuck.

We now define vocabulary for three types of NNTucks, each based on different assumptions of the structure and dimension of $\boldsymbol{Y} \in \mathbb{R}^{L\times C}$.

Definition 1 (Layer independent NNTuck) *A **layer independent NNTuck** is a nonnegative Tucker decomposition where $C = L$ and $\boldsymbol{Y}$ has the constraint $\boldsymbol{Y} = \boldsymbol{I}$.*

Definition 2 (Layer dependent NNTuck) *A **layer dependent NNTuck** is a nonnegative Tucker decomposition where $\boldsymbol{Y}$ has the constraint $C < L$.*

Definition 3 (Layer redundant NNTuck) *A **layer redundant NNTuck** is a nonnegative Tucker decomposition where $C = 1$ and we constrain $\boldsymbol{Y}$ to be the ones vector, $\boldsymbol{Y} = [1, \ldots, 1]^\top$.*

Citation	**Approach**
Schein et al. (2016)	The same model as the nonnegative Tucker decomposition. The factor matrices and core tensor are estimated using an MCMC inference algorithm.
Valles-Catala et al. (2016)	A separate SBM is estimated for each layer.
Stanley et al. (2016)	The model assumes a Bernoulli distribution and K is not fixed across layers. Layers within the same *strata* s are drawn from the same SBM with $\boldsymbol{U}^s := \boldsymbol{V}^s$ constrained to only take on binary values.
Paul and Chen (2016)	$\boldsymbol{U} := \boldsymbol{V}$ constrained to only take on binary values, $\boldsymbol{Y}$ is constrained so that $\boldsymbol{Y} := \boldsymbol{I}$, and the model assumes a Bernoulli distribution.
De Bacco et al. (2017) and Carlen et al. (2022)	$\boldsymbol{Y}$ is constrained so that $\boldsymbol{Y} := \boldsymbol{I}$.
Tarrés-Deulofeu et al. (2019)	$\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{Y}$ have constraint $C = K$ and a new core tensor $\boldsymbol{\mathcal{G}}$ is estimated for each type of link in the network.
Wang and Zeng (2019)	$\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{Y}$ are constrained to only take on binary values.

Table 2: Previous approaches to multilayer SBMs. Excepting the first citation, we write these approaches in relation to the nonnegative Tucker decomposition (NNTuck) which assumes $\boldsymbol{\mathcal{A}} \sim \text{Poisson}(\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y})$ for $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{K \times K \times C}$, $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}_+^{N \times K}$, and $\boldsymbol{Y} \in \mathbb{R}_+^{L \times C}$. For descriptions of our novel contributions situated in this work see Section 2.6 and for more details on the NNTuck see Section 3.

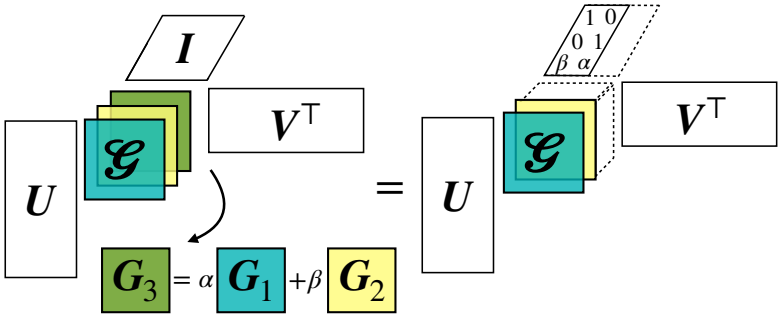


Figure 2: If one or more of the frontal slices of the core tensor are linear combinations of another, there is a *deflation* of the core tensor. In this example, we show how a layer independent NNTuck (left) can be equivalently written as a layer dependent NNTuck (right). This figure shows how layer dependence is stored in the factor matrix $\boldsymbol{Y}$.

3.2 Deflation and Layer Interdependence

In this section we discuss layer dependence in the NNTuck through three examples and discuss how *deflation* of the core tensor allows for latent structure to be identified in the layers of a multilayer network.

Definition 4 (Deflation) *We say there is a deflation of the core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{K\times K\times L}$ of a layer independent NNTuck if there exists a tensor $\boldsymbol{\mathcal{G}}' \in \mathbb{R}_+^{K\times K\times C}$ and a factor matrix $\boldsymbol{Y} \in \mathbb{R}_+^{L\times C}$ for $C < L$ such that,*

$$\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} = \boldsymbol{\mathcal{G}}' \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}. \tag{9}$$

When the core tensor can be deflated the factor matrix $\boldsymbol{Y}$ in the NNTuck captures the interdependence between layers. We examine deflation and the $\boldsymbol{Y}$ factor matrix through the three example model instances below, respectively depicted in Figures 2, 3, and 4.

Example 1 (Linearly dependent core tensor) *For a three-layer adjacency tensor $\boldsymbol{\mathcal{A}} \in \{0,1\}^{N\times N\times 3}$ consider the layer independent NNTuck given by $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \mathbf{I}$ where the frontal slices of core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{K\times K\times 3}$ are as follows:*

$$\boldsymbol{G}_1 = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{bmatrix}, \boldsymbol{G}_2 = \begin{bmatrix} 0.3 & 0.01 \\ 0.01 & 0 \end{bmatrix} \boldsymbol{G}_3 = \begin{bmatrix} 0.35 & 0.105 \\ 0.105 & 0.2 \end{bmatrix}.$$

As may be evident, $\boldsymbol{G}_3$ is a linear combination of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$. Specifically, $\boldsymbol{G}_3 = \boldsymbol{G}_1 + \frac{1}{2}\boldsymbol{G}_2$. In the same sense that a rank-deficient matrix has one or more columns which are a linear combination of others, we can consider the inclusion of $\boldsymbol{G}_3$ in the core tensor redundant. If we have a limited data source from which we are estimating our model, "wasting" information to fit this redundant frontal slice could lead to a less efficiently estimated model. Instead, consider tensor $\boldsymbol{\mathcal{G}}'$ whose frontal slices are $\boldsymbol{G}_1' = \boldsymbol{G}_1$ and $\boldsymbol{G}_2' =$

$\boldsymbol{G}_2$, and define

$$\boldsymbol{Y} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0.5 \end{bmatrix},$$

where the third row contains the respective weights of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ that sum to $\boldsymbol{G}_3$. Then

$$\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \mathbf{I} = \boldsymbol{\mathcal{G}}' \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}.$$

Note that whereas these two models are mathematically equivalent, the *deflated* model allows for latent structure in the layers to be more efficiently identified.

Example 2 (strata SBM) *For $\boldsymbol{\mathcal{A}} \in \mathbb{Z}_{0+}^{N\times N\times 4}$ consider the nonnegative Tucker-2 model given by $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \mathbf{I}$. Assume that the core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}_{+}^{K\times K\times 4}$ has frontal slices $\boldsymbol{G}_3 := \boldsymbol{G}_1$ and $\boldsymbol{G}_4 := \boldsymbol{G}_2$, for the same $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ in the previous example. For $\boldsymbol{\mathcal{G}}' = \mathbb{R}_{+}^{K\times K\times 2}$ with frontal slices $\boldsymbol{G}'_1 = \boldsymbol{G}_1$ and $\boldsymbol{G}'_2 = \boldsymbol{G}_2$ and factor matrix*

$$\boldsymbol{Y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

then $\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \mathbf{I} = \boldsymbol{\mathcal{G}}' \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}$.

The interpretation of this example is that layers 1 and 3 in the multilayer network were drawn from the same SBM, one distinct from that which generated layers 2 and 4. Because node-membership matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are held to be fixed across layers, this means that communities interact with the same rates in layers 1 and 3, although these rates are different from those which determine interaction in layers 2 and 4. This example clusters layers generated from the same SBM. This example is generatively equivalent to the strata SBM (Stanley et al., 2016) if, in the example we fix $\boldsymbol{U} := \boldsymbol{V}$, and if, in the strata SBM K and node-membership are fixed across layers.

Example 3 (repeated SBMs) *For $\boldsymbol{\mathcal{A}} \in \{0,1\}^{N\times N\times 4}$ consider the layer independent NNTuck given by $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \mathbf{I}$. In this example, consider that all of the frontal slices of $\boldsymbol{\mathcal{G}}$ are equal: $\boldsymbol{G}_\ell = \boldsymbol{G}_1$ for $\ell = 1, 2, 3, 4$. Define $\boldsymbol{\mathcal{G}}' \in \mathbb{R}_{+}^{K\times K\times 1} = \boldsymbol{G}_1$ and factor matrix $\boldsymbol{Y} = \mathbf{1} = [1, 1, 1, 1]^\top$. Then $\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \mathbf{I} = \boldsymbol{G}_1 \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \mathbf{Y}$.*

This deflated model is a layer redundant NNTuck and can be interpreted by considering that all layers of the network are different realizations of the *exact same* SBM. That is, the underlying process which is assumed to have generated the structure observed in layer 1 is the same as that which generated the structure observed in all other layers. In this sense, a multilayer network with this multilayer SBM does not need to be represented as a multilayer network. However, see Taylor et al. (2016) for a discussion of the detectibility limit when a multilayer network's layers are generated from a repeated SBM.

The following final example serves as a warning of the limitations of assessing layer interdependence using the NNTuck model.

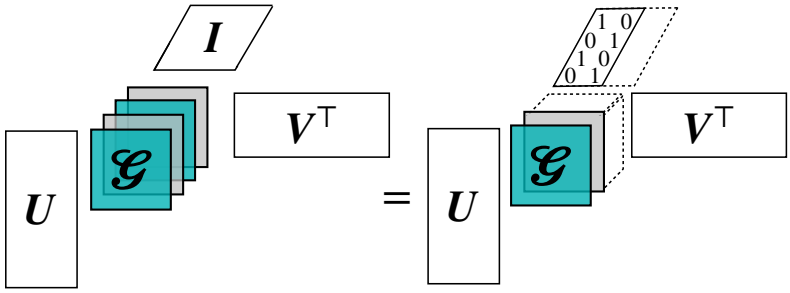


Figure 3: This figure shows how the NNTuck generalizes the strata multilayer stochastic block model (SBM) of Stanley et al. (2016). If, for example, all layers in a multilayer network are drawn from one of two SBMs (with the same $\boldsymbol{U}$ and $\boldsymbol{V}$ across layers), the factor matrix $\boldsymbol{Y}$ has only zeros and ones.

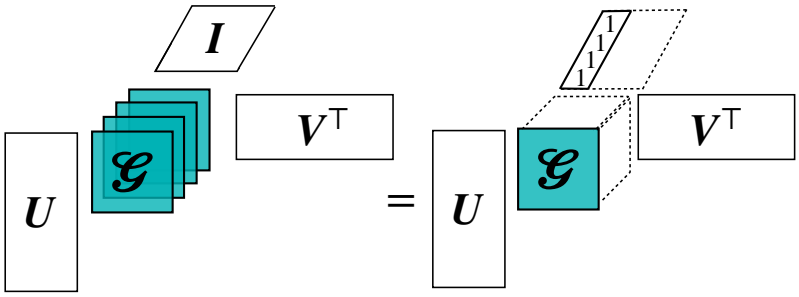


Figure 4: A visualization of a layer redundant NNTuck. This figure shows how NNTuck can model a multilayer network wherein each layer was drawn from the same SBM. In such a case, the factor matrix $\boldsymbol{Y}$ is a vector of all ones and the core tensor $\boldsymbol{\mathcal{G}}$ is of dimension $K \times K \times 1$.

Example 4 (Linear basis as a cause for deflation) *Consider a network for which $K = 2$, $L > 4$, and a layer independent NNTuck with 2×2 affinity matrices $\boldsymbol{G}_\ell$ for $\ell = 1, \ldots, L$ for each layer of the network.*

Note that there is a natural linear algebraic (as opposed to sociological or contextual) reason why this NNTuck can be deflated, or more generally why an estimated model may perform well when $C < L$ without necessarily exhibiting contextually relevant layer dependence. Specifically, *any* 2×2 matrix can be written as a linear combination of the following *bases*:

$$\boldsymbol{B}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \boldsymbol{B}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \boldsymbol{B}_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \boldsymbol{B}_4 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

For example,

$$\boldsymbol{G} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = a\boldsymbol{B}_1 + b\boldsymbol{B}_2 + c\boldsymbol{B}_3 + d\boldsymbol{B}_4.$$

For the NNTuck where all matrices $\boldsymbol{G}_\ell$ have nonnegative entries, coefficients a, b, c, d above will also be nonnegative. Thus for this $K = 2$ case, any core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{2\times 2\times L}$ can be deflated to core tensor $\boldsymbol{\mathcal{B}} \in \mathbb{R}_+^{2\times 2\times 4}$ whose ℓ^{th} frontal slice is $\boldsymbol{B}_\ell$. In this sense, we can only hope to interpret deflation as a characteristic of the network (as opposed to a characteristic of the linear algebra) when $C < K^2$. For an undirected network, where we constrain the frontal slices of the core tensor to be symmetric, this constraint is $C < \frac{K(K+1)}{2}$.

For the empirical examples we consider in the next section (where there are between $L = 7$ and $L = 21$ layers), we will need to consider this issue only in the case where $K = 2, 3, 4$, depending on the network, because $L < K^2$ for all larger K, and thus $C < K^2$ as well. As broader context, for the over 45 datasets provided in De Domenico's multilayer network database (De Domenico, 2022), only two datasets have more than 16 layers. This, however, is not always the case, and especially not so when considering temporal multilayer networks wherein a network is captured at many time steps. In such situations, one must bear in mind the constraint of $C < K^2$ for the purposes of interpreting the NNTuck.

Remark 5 (Relationship to MULTITENSOR (MT)) *If we collect the affinity matrices $\boldsymbol{G}_\ell$ from the MT model to be the frontal slices of tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{K\times K\times L}$, then (7) is equivalent to*

$$\boldsymbol{\mathcal{A}} \sim \textit{Poisson}(\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}). \tag{10}$$

Note that $\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \mathbf{I}$. This model is what we define as a layer independent NNTuck in Theorem 1 above and is sometimes called a Tucker-2 decomposition (see Kolda and Bader, 2009). Since all factors are nonnegative, the MT model seeks to find the nonnegative Tucker-2 decomposition by maximizing the log-likelihood through EM. Given that the interpretation of the $\boldsymbol{Y}$ factor matrix is that it describes layer communities in the network, constraining $\boldsymbol{Y} = \boldsymbol{I}$ as is done in MT assumes that each layer of the multilayer network was drawn from a distinct SBM, albeit with common membership matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. That is, MT assumes that there is no latent structure in the layers of the network.

3.3 Algorithmic Approach

Kim and Choi (2007) extend the multiplicative updates for NMF from Lee and Seung (2000) to the nonnegative Tucker decomposition for minimizing both KL-divergence and Frobenius

loss. We reproduce the updates for minimizing KL-divergence in Algorithm 1. The updates in Kim and Choi (2007) are written for a general, n-th order tensor, so we rewrite them here for a 3rd order tensor in a setting wherein some data is masked as specified by a masking tensor $\boldsymbol{\mathcal{M}} \in \{0,1\}^{N\times N\times L}$. Note that if the data is not masked, the all-ones masking tensor $\boldsymbol{\mathcal{M}} = 1^{N\times N\times L}$ recovers the original multiplicative updates. As we'll see in Section 5, the masking tensor allows for cross-validation wherein the NNTuck is only estimated from a portion of the network. Note that these updates are done sequentially, not in parallel. See the URL in Appendix A for a python implementation.

Algorithm 1 Multiplicative Updates for minimizing KL-Divergence in the NNTuck (Kim and Choi, 2007)

Input: $\boldsymbol{\mathcal{A}}, K, C$, `Symmetric`, `Masked`, $\boldsymbol{\mathcal{M}}$, `Independent`, `Redundant`
Initialize $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}_+^{N\times K}, \boldsymbol{Y} \in \mathbb{R}_+^{L\times C}$, and $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{K\times K\times C}$ to have random, nonnegative entries. Initialize $\hat{\boldsymbol{\mathcal{A}}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}$.
if `Symmetric`: $\boldsymbol{V} \leftarrow \boldsymbol{U}$, $\boldsymbol{G}_\ell \leftarrow \boldsymbol{G}_\ell^\top \boldsymbol{G}_\ell$ for $\ell = 1, \ldots, C$, and skip each $\boldsymbol{V}$ update step below.
if `Independent`: $\boldsymbol{Y} \leftarrow \boldsymbol{I}$ and skip each $\boldsymbol{Y}$ update step below.
if `Redundant`: $\boldsymbol{Y} \leftarrow$ `ones(C)` and skip each $\boldsymbol{Y}$ update step below.
if not `Masked`: $\boldsymbol{\mathcal{M}} = 1^{N\times N\times L}$

while $\frac{KL(\boldsymbol{\mathcal{A}}||\hat{\boldsymbol{\mathcal{A}}}_t) - KL(\boldsymbol{\mathcal{A}}||\hat{\boldsymbol{\mathcal{A}}}_{t-1})}{KL(\boldsymbol{\mathcal{A}}||\hat{\boldsymbol{\mathcal{A}}}_t)} <$ `rel_tol`:

$$\boldsymbol{U} \leftarrow \boldsymbol{U} \circ \frac{[\boldsymbol{M}_{(1)} \circ \boldsymbol{A}_{(1)} / \hat{\boldsymbol{A}}_{(1)}][\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)}^\top}{\boldsymbol{M}_{(1)}[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)}^\top}$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} \circ \frac{[\boldsymbol{M}_{(2)} \circ \boldsymbol{A}_{(2)} / \hat{\boldsymbol{A}}_{(2)}][\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)}^\top}{\boldsymbol{M}_{(2)}[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)}^\top}$$

$$\boldsymbol{Y} \leftarrow \boldsymbol{Y} \circ \frac{[\boldsymbol{M}_{(3)} \circ \boldsymbol{A}_{(3)} / \hat{\boldsymbol{A}}_{(3)}][\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}]_{(3)}^\top}{\boldsymbol{M}_{(3)}[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}]_{(3)}^\top}$$

$$\boldsymbol{\mathcal{G}} \leftarrow \boldsymbol{\mathcal{G}} \circ \frac{[\boldsymbol{\mathcal{M}} \circ \boldsymbol{\mathcal{A}} / \hat{\boldsymbol{\mathcal{A}}}] \times_1 \boldsymbol{U}^\top \times_2 \boldsymbol{V}^\top \times_3 \boldsymbol{Y}^\top}{\boldsymbol{\mathcal{M}} \times_1 \boldsymbol{U}^\top \times_2 \boldsymbol{V}^\top \times_3 \boldsymbol{Y}^\top}$$

$$\hat{\boldsymbol{\mathcal{A}}} \leftarrow \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}$$

Return $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{Y}, \boldsymbol{\mathcal{G}}$.

Because these updates are derived from the multiplicative updates for NMF from Lee and Seung (2000), they come with guaranteed monotonic convergence to a local minima. In practice, we declare that the algorithm has found a local minima if the KL-divergence has not decreased by more than a relative tolerance of 10^{-5} in ten steps. For the case of an undirected network, we initialize the core tensor to have symmetric frontal slices ($\boldsymbol{G}_\ell = \boldsymbol{G}_\ell^\top$), and initialize and fix $\boldsymbol{U} = \boldsymbol{V}$ throughout the updates. We then follow the multiplicative updates above by only making updates to $\boldsymbol{U}, \boldsymbol{Y}$, and $\boldsymbol{\mathcal{G}}$. Doing so maintains the guaranteed monotonic convergence to a local minima while preserving the symmetric

structure in $\boldsymbol{\mathcal{G}}$, and ensures the constraint for the undirected case that $\boldsymbol{U} = \boldsymbol{V}$. A proof of the following Proposition appears in Appendix B.

Proposition 6 *Determining factor matrices $\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{Y}$ and the core tensor $\boldsymbol{\mathcal{G}}$ in the NNTuck by maximizing the log-likelihood using expectation maximization (EM) is equivalent to using the multiplicative updates given in Kim and Choi (2007) to minimize KL-divergence.*

The significance of this proposition is noticing that not only is minimizing KL-divergence equivalent to maximizing log-likelihood, but also that the *algorithm* by which to find a local minimum of the KL-divergence is the exact same as that to find a local maximum of the log-likelihood. Moreover, using EM to maximize the log-likelihood of observing $\boldsymbol{\mathcal{A}}$ under the De Bacco et al. (2017) MULTITENSOR (MT) model is equivalent to minimizing the KL-divergence between $\boldsymbol{\mathcal{A}}$ and a layer independent NNTuck. That is, the EM steps given in De Bacco et al. (2017) are equivalent to the multiplicative updates in Kim and Choi (2007), where at initialization $\boldsymbol{Y} = \boldsymbol{I}$ is fixed and at each step $\boldsymbol{Y}$ is not updated. The algorithmic equivalence between EM for a Poisson model and multiplicative updates has been noted for NMF in Févotte and Cemgil (2009) and for the CP decomposition in Chi and Kolda (2012).

Algorithmic limitations It is important to emphasize that the the KL-divergence given by (6) (and thus the log-likelihood of observing $\boldsymbol{\mathcal{A}}$ under (8)) is non-convex. Therefore, although the multiplicative updates discussed above guarantee monotonic convergence, it is only to local optima. In practice, we use a multistart approach: run the algorithm multiple times with different initial conditions and select the NNTuck with the maximal log-likelihood over these runs. See Appendix A for details in choosing the number of random initializations. Going forward, we use hat notation to denote the NNTuck factors and core tensor estimated by Algorithm 1 using a multistart approach.

Although not strictly algorithmic, as we note in the next section, the nonnegative Tucker decomposition is non-identifiable. However, we note that in practice when we compare multiple decompositions of the same multilayer network, there is little to no interpretable difference in the factors associated with the estimation with the maximal log-likelihood over 20 random initializations. Although recent work by Sun and Huang (2023) has proposed conditions for ensuring identifiability, implementing and altering these conditions in the context of multilayer networks, is subject for future work. The non-identifiability of the NNTuck is one reason why, as we'll see in the next section, we emphasize the use of the *split likelihood ratio test* Wasserman et al. (2020) to test for interdependent structure in multilayer networks.

The algorithmic details and implementation of the NNTuck (outside of its application to multilayer networks) is a rich field with many future work directions. For example, future work could explore an evaluation of alternative optimization methods for nonnegative matrix and tensor factorizations, including mirror descent (Hien and Gillis, 2021), projected gradient descent (Cichocki and Zdunek, 2007), and stochastic gradient descent (Kasai, 2018). Furthermore, Chi and Kolda (2012) propose a related algorithm for nonnegative Poisson CP decomposition using multiplicative updates and discuss conditions under which the algorithm converges to KKT points. A similar analysis of the nonnegative Tucker decomposition would be intriguing but is again outside the scope of this work.

4. Statistical Tests for Validating Layer Interdependence

In this section we introduce formal definitions of *layer interdependence* by defining corresponding likelihood ratio tests (LRTs). We conclude with a presentation of three methods by which to interpret $\hat{\boldsymbol{Y}}$ of an NNTuck estimated for an empirical multilayer network.

4.1 Layer Interdependence and Likelihood Ratio Tests

Likelihood ratio tests can be used to assess the performance of two models, where one model is nested within the other. In the context of evaluating different NNTuck models we compare the layer independent NNTuck to the nested models of layer dependent NNTucks or a layer redundant NNTuck. The null hypothesis of the LRT is that the two models fit the data equally well, and the alternative hypothesis is that the richer model fits the data significantly better. If the resulting p-value rejects the null hypothesis, then the full model should be used. Otherwise, the nested model should be used. We use this framework to define three tests for multilayer networks.

Definition 7 (Layer independence) *For a multilayer network let model I be the layer independent NNTuck and let model II be the layer dependent NNTuck. A multilayer network has* ***layer independence*** *at level α if the likelihood ratio test with $(L-C)K^2 - LC$ degrees of freedom is significant at level α.*

Definition 8 (Layer dependence) *A multilayer network has* ***layer dependence*** *at level α if the LRT described above is not significant at level α for a pre-specified C.*

Definition 9 (Layer redundance) *A multilayer network has* ***layer redundance*** *at level α, if the LRT comparing the layer redundant NNTuck to the $C = 2$ layer dependent NNTuck with $K^2 + 2L$ degrees of freedom is not significant at level α.*

To use these LRTs, one must determine how many parameters are in the full model and how many are in the nested model. For example, to find the difference in the number of parameters between the layer dependent and independent NNTuck, consider that the layer independent NNTuck has $N \times K$ parameters in $\boldsymbol{U}$ and $N \times K$ parameters in $\boldsymbol{V}$. There are $K \times K \times L$ parameters in $\boldsymbol{\mathcal{G}}$ and no free parameters in $\boldsymbol{Y}$ because it is fixed. The layer dependent NNTuck has $N \times K$ parameters in each of $\boldsymbol{U}$ and $\boldsymbol{V}$, $L \times C$ parameters in $\boldsymbol{Y}$, and $K \times K \times C$ parameters in $\boldsymbol{\mathcal{G}}$. Thus the difference in parameters between both models is $2NK + K^2L - 2NK - LC - K^2C = (L-C)K^2 - LC$. Likewise, when comparing two layer dependent NNTucks with dimensions C_f and C_n for $C_n < C_f$ and fixed K, there is a difference of $(L + K^2)(C_f - C_n)$ parameters between the two models. When comparing the layer redundant NNTuck nested under the $C = 2$ layer dependent NNTuck, the difference in number of parameters is $K^2 + 2L$.

It is important to note that the theory underlying the likelihood ratio test, Wilks' theorem (Wilks, 1938), necessarily depends on (i) the maximum likelihood being reached and (ii) the model being identifiable. These are conditions we cannot guarantee in our problem context. Moreover, we propose this method for determining layer interdependence constrained to the classes of models for which the difference in the degrees of freedom $d = (L-C)K^2 - LC > 0$. Because of the non-identifiability, the way by which these models

are nested is nuanced, and thus this inequality is not always true for certain values of L, K, and C. Furthermore, Wilks' theorem gives *asymptotic* analysis of how the difference between two likelihoods approaches a χ^2 distribution with degrees of freedom given by the difference in parameters. To explore how the number of samples—the number of nodes and layers, in the context of a multilayer network—impacts the power of these LRTs, we conduct a numerical experiment in Appendix F.

To address both of these issues, we also utilize the *split-LRT*, developed by Wasserman et al. (2020), which requires no regularity conditions. The split-LRT, however, still requires that the estimation of the nested model corresponds to the global maximum of the log-likelihood. Algorithm 1 only guarantees convergence to a local maxima, so for comparing models using both the standard and the split LRT, we use the NNTuck corresponding to the highest log-likelihood over multiple initializations of Algorithm 1. See Figure 11 in Appendix A to see how the maximal log-likelihood achieved by Algorithm 1 varies across multiple initializations, and see Appendix G for more details on split-LRT. For the datasets we discuss below, the layer independence, dependence, and redundancy tests only differ for one dataset when comparing the regular LRT to the split-LRT. This difference is consistent with the fact that the split-LRT is lower powered than the regular LRT. The low power of the split-LRT may be intensified in the context of the NNTuck by the presence of *nuisance parameters*, wherein although the LRT is testing for structure in $\boldsymbol{Y}$, the other factor matrices and core tensor must be estimated as well. In Tse and Davison (2022), Strieder and Drton (2022), and Spector et al. (2023), the authors propose methods for improving the power of the split-LRT in the presence of nuisance parameters, including suggestions for how to more optimally split the data. An analysis considering how to extend their suggestions to the setting of three-way tensor data is beyond the scope of this work. Alternative LRTs for latent variable models have also been proposed (see Chen et al., 2020), and address other common issues that arise when using the LRT to compare latent variable models.

4.2 Layer Interdependence in an Estimated NNTuck

If the layer dependence test determines that an empirical multilayer network has dependent layers, it is useful to investigate *how* they are related. In the examples in Section 3.2 above, the frontal slices of the deflated core tensor correspond exactly to the affinity matrix of one or more of the layers. As an example, consider the frontal slices of the deflated NNTuck in Figure 2. These frontal slices are the affinity matrices for the first and second layer of the multilayer network (beyond the color coding in the example, we can also see this in the first two rows of the $\boldsymbol{Y}$ factor matrix, which are $[1, 0]$ and $[0, 1]$, respectively). For an NNTuck estimated for an empirical multilayer network there is no constraint such that this must be true. As such, one must use certain heuristics to appropriately interpret $\hat{\boldsymbol{Y}}$ estimated from empirical data.

The first approach is to row-normalize $\hat{\boldsymbol{Y}}$ such that $\hat{\mathbf{y}}_\ell^{(1)} = \hat{\mathbf{y}}_\ell / \|\hat{\mathbf{y}}_\ell\|_1$ and inspect the rows of $\hat{\boldsymbol{Y}}^{(1)}$ relative to one another. The second approach is to row-normalize $\hat{\boldsymbol{Y}}$ such that $\hat{\mathbf{y}}_\ell^{(2)} = \hat{\mathbf{y}}_\ell / \|\hat{\mathbf{y}}_\ell\|_2$ and inspect the entries of similarity matrix given by $\hat{\boldsymbol{Y}}^{(2)}\hat{\boldsymbol{Y}}^{(2)\top}$. The third approach uses *reference layer bases*. In this approach, C reference layers are chosen, $\hat{\boldsymbol{\mathcal{G}}}$ is rewritten in the linear bases of those reference layers' affinity matrices, and corresponding $\hat{\boldsymbol{Y}}^*$ is defined in relation to the new core tensor. This last approach has the added benefit

of interpreting each layer's dependence *with respect to* the C reference layers. For specifics on this process and guidance on choosing the reference layers, see Appendix E.

5. Model Selection Through Cross-Validation

In this section we discuss the use of cross-validation in this work and define two link prediction tasks for tensors. We emphasize that we are more interested in how the cross-validation highlights interesting model choices than by the actual predictive performance of NNTuck in these link prediction tasks. Statistical factor models of networks are generally not competitive with machine learning classifiers that use even simple topological features (see, e.g., Clauset et al., 2008; Liben-Nowell and Kleinberg, 2007; Ghasemian et al., 2020). As such, the absolute performance here should not be considered a metric of primary interest, but as a means of comparative inspection. That said, recall that in the following link prediction tasks the layer independent NNTuck is equivalent to MULTITENSOR from De Bacco et al. (2017); to compare the performance of NNTuck to other link prediction methods, see De Bacco et al. (2017).

The construction of the cross-validation approach is as follows. For each link prediction task we construct five different masking tensors and estimate a model based on only observed entries of the data tensor. We select the NNTuck with the highest test set log-likelihood from 50 different runs of the multiplicative updates algorithm with random initializations. Then, test-AUC is averaged across the five different maskings. This process is repeated for varying dimensions (K, C) in the NNTuck.

We define the link prediction tasks via the structure of their *masking tensors* $\boldsymbol{\mathcal{M}} \in \{0,1\}^{N\times N\times L}$ where $M_{ij\ell} = 0$ indicates that the presence or absence of an edge between nodes i and j in layer ℓ is missing, and $M_{ij\ell} = 1$ otherwise. In undirected networks we enforce $M_{ij\ell} = M_{ji\ell}$ for both link-prediction tasks.

Independent link prediction In this link prediction task masking is irrespective of layer. That is, we assume that for b-fold cross-validation, elements in the tensor are missing with uniform and independent probability $1/b$. Specifically, missing entry (i, j) in layer k does not imply that entry (i, j) is missing in all layers ($M_{ijk} = 0 \not\Rightarrow M_{ij\ell} = 0$ for $\ell \neq k$).

Tubular link prediction In this link prediction task edges are always observed or missing *across all layers*. That is, we assume that for b-fold cross-validation, tubes $(i, j, \cdot)$ in the tensor are missing with uniform probability $1/b$ (see Figure 13 for a visualization of tensor tube fibers). Specifically, missing link (i, j) in layer k *does* imply that link (i, j) is missing in all layers ($M_{ijk} = 0 \Rightarrow M_{ij\ell} = 0$, $\forall \ell$). We motivate this tubular task by commenting that independent link prediction is often "too easy," in the sense that if many layers are dependent then missing elements are much easier to impute when other elements from the same tube are available. Tubular link prediction captures realistic settings where one knows nothing at all about the relationship between two units i and j in any layer.

Given the structure of an adjacency tensor, there are (at least) two other link prediction tasks which are representative of true missingness patterns in data: one in which an entire horizontal slice of data is missing ($M_{ijk} = 0 \Rightarrow M_{ip\ell} = 0$ for all p, ℓ), and one in which an entire lateral slice of data is missing ($M_{ijk} = 0 \Rightarrow M_{rj\ell} = 0$ for all r, ℓ). We limit our

scope to the independent and tubular link prediction tasks above, but mention these as potentially interesting missingness patterns in the context of multilayer networks.

6. Application of the NNTuck to Synthetic and Empirical Networks

In this section we use the cross-validation tools discussed in Section 5, the layer dependence tests developed in Section 4.1, and the $\boldsymbol{Y}$ interpretability heuristics from Section 4.2 to use the NNTuck in application. In Section 6.1 we generate a synthetic network example to exhibit the interpretability of the $\boldsymbol{Y}$ factor matrix when K and C are known. In Section 6.2 for each empirical network presented we carry out the following steps: (i) use a cross-validation approach to determine model hyper-parameter choices K and C, (ii) use likelihood ratio tests with this (K, C) pair to determine layer independence, redundance, or dependence, and (iii) if the network is layer dependent at level α, examine the $\hat{\boldsymbol{Y}}$ factor matrix.

6.1 Synthetic network examples

In this section we define two different synthetic networks and inspect their estimated $\hat{\boldsymbol{Y}}$ factor matrices. For both examples, we let $N = 200$, $K = 2$, $L = 4$, and define affinity matrices

$$\boldsymbol{G}_1 = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{bmatrix} \text{ and } \boldsymbol{G}_3 = \begin{bmatrix} 0.3 & 0.01 \\ 0.01 & 0 \end{bmatrix}.$$

We set the affinity matrices $\boldsymbol{G}_2$ and $\boldsymbol{G}_4$ to be linear combinations of the above affinity matrices,

$$\boldsymbol{G}_2 = a\boldsymbol{G}_1 + b\boldsymbol{G}_3 \text{ and } \boldsymbol{G}_4 = c\boldsymbol{G}_1 + d\boldsymbol{G}_3,$$

for different values of a, b, c, and d between the two examples.

In both examples, we generate a multilayer network from an SBM which assumes that an edge between nodes i and j in layer ℓ is drawn from a Poisson distribution with mean $\mathbf{u}_i \boldsymbol{G}_\ell \mathbf{v}_j^T$. We set $\mathbf{u}_i = \mathbf{v}_i$ and assign 100 nodes to the first group ($\mathbf{u}_i = [1, 0]$) and 100 nodes to the second group ($\mathbf{u}_i = [0, 1]$). Generating these synthetic networks in this way is equivalent to drawing them from $\boldsymbol{\mathcal{A}} \sim \text{Poisson}(\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y})$ for

$$\boldsymbol{Y} = \begin{bmatrix} 1 & 0 \\ a & b \\ 0 & 1 \\ c & d \end{bmatrix}$$

and $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{K \times K \times 2}$ with first and second frontal slices $\boldsymbol{G}_1$ and $\boldsymbol{G}_3$, respectively. For the first network we define $a = 0.5$, $b = 0.5$, $c = 0.1$, and $d = 0.9$, and for the second we let $a = 1$, $b = 0$, $c = 0$, and $d = 1$. This second synthetic network is the strata example depicted in Figure 3 and discussed in Example 2. As an aside, note that whereas in these two networks the entries of the rows of $\boldsymbol{Y}$ sum to one, this need not be the case. Actually, by allowing the rows of $\boldsymbol{Y}$ to be unnormalized we can account for heterogeneous degree distributions *across layers*, just as the degree-corrected single layer SBM in Karrer and Newman (2011) accounts for heterogeneous degree distributions across nodes.

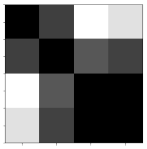
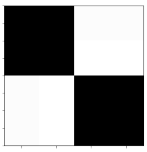
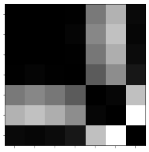
$\boldsymbol{Y}$	$\hat{\boldsymbol{Y}}$	$\hat{\boldsymbol{Y}}^{(1)}$	$\hat{\boldsymbol{Y}}^{(2)}\hat{\boldsymbol{Y}}^{(2)\mathsf{T}}$	$\hat{\boldsymbol{Y}}^{*}$
$\begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \\ 0.1 & 0.9 \end{bmatrix}$	[[0.22616 0.00008] [0.11345 0.0996] [0. 0.18084] [0.02043 0.17165]]	[[0.99967 0.00033] [0.53251 0.46749] [0. 1.] [0.10635 0.89365]]		[[1. 0.] [0.50165 0.55057] [0. 1.] [0.09032 0.94914]]
$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$	[[0.00389 0.2282] [0.00061 0.22871] [0.19255 0.00001] [0.19034 0.]]	[[0.01678 0.98322] [0.00264 0.99736] [0.99997 0.00003] [0.99999 0.00001]]		[[1. 0.] [1.00222 -0.01712] [0. 1.] [-0.00002 0.98853]]
Gossip village #48 $K = 4,\ C = 2$	[[0.00167 0.00613] [0.00135 0.00611] [0.00193 0.00678] [0.00276 0.00669] [0.00396 0.00185] [0.00402 0.001] [0.00007 0.00311]]	[[0.21378 0.78622] [0.18067 0.81933] [0.22132 0.77868] [0.2919 0.7081] [0.68134 0.31866] [0.80135 0.19865] [0.02254 0.97746]]		[[0.17075 0.82925] [0.1388 0.8612] [0.17817 0.82183] [0.25064 0.74936] [0.77642 0.22358] [1. 0.] [0. 1.]]

Figure 5: We reproduce the results of the methods for interpreting $\hat{\boldsymbol{Y}}$ in the NNTuck of the first and second synthetic network described above as well as for the 48th village from Banerjee et al. (2019) (labelled "Gossip village 48" in Figure 9). For the synthetic networks, $\boldsymbol{Y}$ is the true factor matrix from which the network was generated. For all three, $\hat{\boldsymbol{Y}}$ has been estimated from the NNTuck with the highest log-likelihood over 20 runs with different random initializations, $\hat{\boldsymbol{Y}}^{(1)}$ has been normalized so that the entries of each row sum to one, $\hat{\boldsymbol{Y}}^{(2)}$ has been normalized so that each row has unit 2-norm. For the synthetic networks, $\hat{\boldsymbol{Y}}^{*}$ is the resulting factor matrix after rewriting $\boldsymbol{\mathcal{G}}$ in the basis of layers 1 and 3 (a process which is described in detail in Appendix C). Note that in the synthetic examples, all methods for interpreting $\hat{\boldsymbol{Y}}$, including simply inspecting $\hat{\boldsymbol{Y}}$, accurately represent how the layers of the network are related to one another. Specifically, note how $\hat{\boldsymbol{Y}}^{*}$ almost exactly recovers the ground truth of how the layers are interdependent. Focusing on $\hat{\boldsymbol{Y}}^{*}$ matrix for the gossip village, the two reference layers chosen are "Who asks you for advice?" and "Who are your relatives?", where the remaining layers can be understood in terms of a linear combination of these.

For both networks we estimate the NNTuck with $C = 2$ and $K = 2$ and report the NNTuck with the highest log-likelihood over 20 runs with different random initializations. We threshold the values in the resulting membership matrices $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$ to reflect the hard membership of the generative model. Node membership is *exactly* recovered for both networks and thus we focus our attention on interpreting $\hat{\boldsymbol{Y}}$. In both networks, $\hat{\boldsymbol{Y}}$ recovers the structural dependence between layers and we report the results of all three approaches for interpreting $\hat{\boldsymbol{Y}}$ in Figure 5.

6.2 Empirical Multilayer Networks

In this section we use the NNTuck and the tools developed thus far to study several empirical datasets: the cognitive social structure dataset from Krackhardt (1987); a biological multilayer network from Larremore et al. (2013); a social support multilayer network from Banerjee et al. (2013); and 112 other multilayer social support networks from Banerjee et al. (2013, 2019). In Table 3 we include the results from the LRTs for a subset of these empirical networks and a synthetic network from Section 6.1. Notably, we conclude that the Malaria multilayer network has layer independence at level $\alpha = 0.05$, whereas and all of the other datasets have either layer redundance or layer dependence at the same level α.

Note that for the cross-validation tasks in the following subsections, we report the average test AUC across 50 different random initializations for each combination of K and C. We vary K from 2 to 12 in the Krackhardt multilayer network, and from 2 to 20 in the Malaria and Village multilayer networks. See Appendix D for a discussion of how we vary K and to see how increasing K to larger values does not impact our model selection. For the LRT in each application we select the NNTuck (for prespecified (K, C)) with the highest log-likelihood across 20 random initializations. See Appendix A for computational experiments testing the variation in maximal log likelihood as a function of the number of random restarts.

Dataset	Test	standard LRT	split-LRT
Malaria	H_0: Redundant H_1: $C = 2$	p `<1e-16` reject H_0	reject H_0
	H_0: Dependent $K = 5, C = 2$ H_1: Independent	p `<1e-16` reject H_0	reject H_0
Village 0	H_0: Redundant H_1: $C = 2$	p `1.0` fail to reject H_0	fail to reject H_0
	H_0: Dependent $K = 5, C = 2$ H_1: Independent	p `1.0` fail to reject H_0	fail to reject H_0
Krackhardt	H_0: Redundant H_1: $C = 2$	reject H_0	reject H_0
	H_0: Dependent $K = 3, C = 4$ H_1: Independent	p `0.451` fail to reject H_0	fail to reject H_0
Synthetic	H_0: Redundant H_1: Dependent $C = 2$	p `<1e-16` reject H_0	reject H_0
	H_0: Dependent $K = C = 2$ H_1: Independent	p `1.0` fail to reject H_0	fail to reject H_0
Gossip 48	H_0: Redundant H_1: Dependent $C = 2$	p `<1e-16` reject H_0	fail to reject H_0
	H_0: Dependent $K = 4, C = 2$ H_1: Independent	p `1.0` fail to reject H_0	fail to reject H_0

Table 3: The standard and split-LRT determinations for all datasets explored in the following sections. For the standard LRT, the p-values for each test are also reported. The Village 0 support system network is determined to be layer redundant, the Malaria network is determined to have layer independence, and the Krackhardt network is determined to have layer dependence. The layer redundant test for Gossip Village 48 is the only one wherein the standard and split LRTs do not agree, and the difference is consistent with the split-LRT being lower powered than the standard LRT. Both the standard and split-LRT determine that Gossip Village 48 is layer dependent at level $\alpha = 0.05$, and we explore this empirical $\hat{\boldsymbol{Y}}$ in Figure 5.

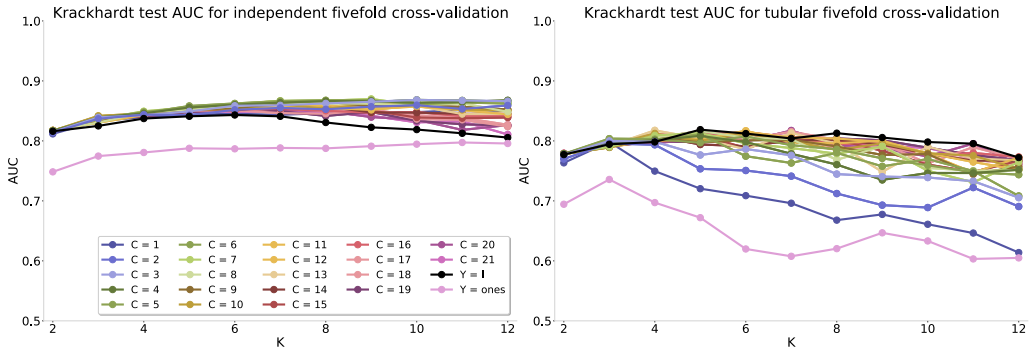


Figure 6: NNTuck performance on independent (left) and tubular (right) link prediction tasks with varying latent dimensions K and C for Krackhardt's CSS multilayer network. Whereas the layer dependent NNTuck with $C < L$ has a higher test-AUC in the independent task, the layer independent NNTuck generally performs as well as the other models in the tubular task. Recall that in this figure, as well as in Fig. 7 and Fig. 8, the layer independent NNTuck is equivalent to MULTITENSOR (De Bacco et al., 2017).

6.2.1 Krackhardt's Cognitive Social Structures

The Cognitive Social Structures work from Krackhardt (1987) surveys 21 people in the management team at a tech firm on their perception of the advice network within the management team. Each of the 21 people were asked to answer the question "Who would X go to for help or advice at work?" followed by a list of the 21 management employees (including themselves). The resulting $21 \times 21 \times 21$ multilayer network is what Krackhardt referred to as a cognitive social structure (CSS), where each layer ℓ represents person ℓ's perception of who receives advice from whom in the network. The adjacency matrices for this advice CSS were transcribed from the original paper for this work, and can be accessed on GitHub (see Aguiar, 2021) (the CSS for the friendship network is different and can be accessed in the R package `cssTools`, see Yenigun et al. (2016)).

Interestingly, the cross-validation observations are different for each link prediction task. We observe a higher test-AUC associated with the layer dependent NNTuck in the independent link prediction task, and becomes more pronounced as K increases. In the tubular link prediction task, however, the layer independent and layer dependent NNTucks have a similar test-AUC for nearly all values of K. In both link prediction tasks we observe: variation in test-AUC for different values of K and C; the layer redundant NNTuck has a lower test-AUC than the layer dependent or layer independent NNTucks; neither the independent nor tubular link prediction task is obviously harder than the other; and the observations from the independent link prediction task are not the same as those from the tubular link prediction task. One possible source of this difference is that the tubular link prediction task is the more difficult one, when compared to the independent task, and thus the results from this task are more representative of a model's performance.

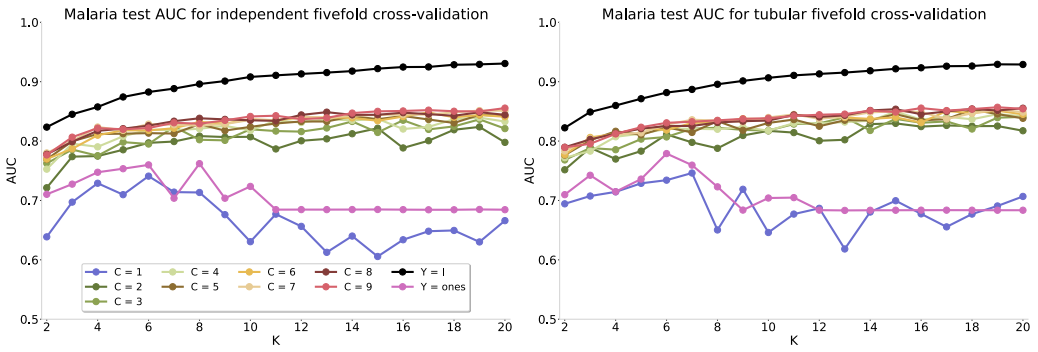


Figure 7: The test-AUC from the independent and tubular link prediction tasks in the Malaria multilayer network. The layer independent NNTuck always results in a higher test-AUC than when allowing for layer dependence or layer redundance.

Based on these results, we choose $K = 3$ and $C = 4$ for the corresponding layer independence, redundance, and dependence tests and determine that whereas the network is not layer redundant, it is layer dependent at significance level $\alpha = 0.05$ (see Table 3 for details). For the sake of brevity (and due to its size (21×4)), we do not interpret the $\hat{\boldsymbol{Y}}$ matrix here. We do note that observing layer dependence in this multilayer network suggests there is latent and shared structure in the various social network perceptions in this company. For a further exploration of this finding and interpretation in the context of theories of social network perceptions, see Aguiar and Ugander (2024).

6.2.2 Malaria Data

This biological network was originally studied in Larremore et al. (2013). The undirected network consists of $N = 307$ malaria parasite virulence genes connected across $L = 9$ layers. Two genes are connected if they share a genetic substring of a significant length. Each layer corresponds to a different *Highly Variable Region* on the genes. For more information on the framework or motivating underlying biology, see Larremore et al. (2013).

In this network the test-AUC of the layer independent NNTuck is always higher than the test-AUC of either the layer dependent or layer redundant NNTuck. This performance difference indicates that the core tensor cannot be deflated without losing important information about the network's layers. Interestingly, we observe that the layer dependent NNTuck with $C = 9$ does not perform as well as the layer independent NNTuck, even though the core tensor has the same dimension in both models. While this observation may be an artifact of the underlying optimization landscape, we do not fully understand the implications or causes and it is an interesting topic for future work. Finally, we do not observe a gap in test-AUC between the independent and tubular link-prediction tasks: predicting a missing link with information about that link in other layers is just as difficult as predicting a missing link with no other information about that link in *any* layer.

We therefore determine that an appropriate model choice is a layer independent NNTuck with $K = 5$ and find that this network is layer independent at significance level $\alpha = 0.05$.

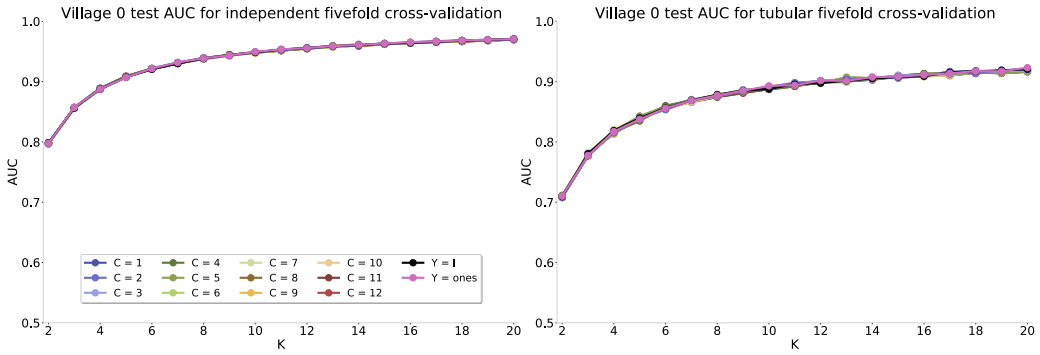


Figure 8: The test-AUC from the independent and tubular link prediction tasks for the Village 0 multilayer network. In both tasks, both the layer dependent and layer redundant NNTucks perform just as well as the layer independent NNTuck in terms of test-AUC.

Therefore, $\hat{\boldsymbol{Y}} = \boldsymbol{I}$ and thus does not need to be interpreted. Finding evidence of layer independence in this multilayer network is supported by a biological explanation (Larremore et al., 2013) and by the findings discussed in De Bacco et al. (2017), namely that the diversity of Malaria genes helps them to evade the immune system.

6.2.3 Village Social Support Network

The social multilayer network we consider contains different types of social interaction within a village in Karnataka, India, one of 43 *microvinance villages* from Banerjee et al. (2013). We arbitrarily selected the first of the 43 villages and will henceforth refer to this village as "Village 0". The directed network consists of $N = 843$ individuals across $L = 12$ layers. One individual is connected to another if the first indicated that they would interact in a specified way with the second. Each layer corresponds to a different type of social interaction (e.g., "Who are the people who give you advice?" and "Who are your kin?". See Appendix H for a full list of the questions.). For more information about the networks, survey instruments, or context of this data, see Banerjee et al. (2013).

Cross validation results for the Village 0 network are shown in Figure 8. There is a slight gap in test-AUC across the two link-prediction tasks for this dataset, where the tubular link-prediction task is more difficult than the independent link-prediction task. However, in both tasks we observe that the layer redundant NNTuck and the layer dependent NNTucks (for all C) perform just as well as the layer independent NNTuck in terms of test-AUC.

The layer redundancy test confirms that this network is layer redundant at significance level $\alpha = 0.05$, consistent with the notion that the 12 layer network may indeed be such that all layer models are drawn from the same SBM, as we saw in Example 3. Considering the efforts made to collect data on these 12 different social support systems, this observation is surprising. One would expect that the distinct questions generating each layer of the network capture new information about the social network. This observation suggests otherwise, at least for the social structures that are well-modeled by stochastic block models.

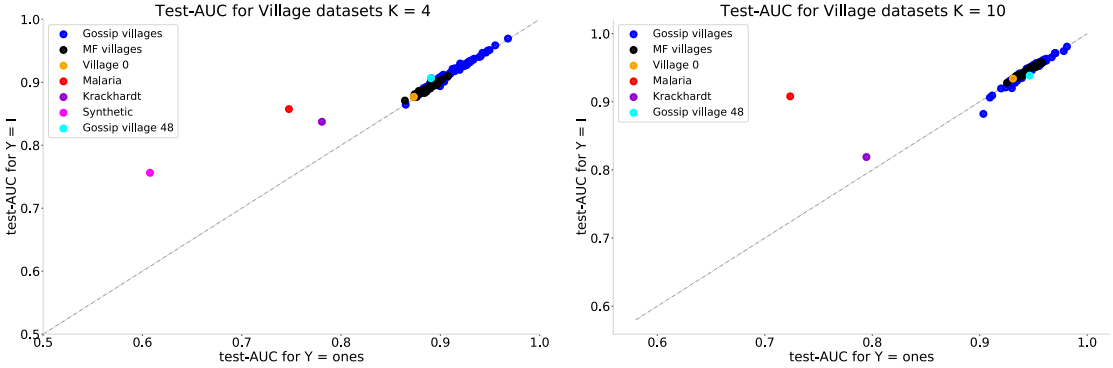


Figure 9: Comparing the test-AUC from the layer redundant NNTuck to that from the layer independent NNTuck with $K = 4$ (left) and $K = 10$ (right) for 113 different village multilayer networks from Banerjee et al. (2013) and Banerjee et al. (2019), the malaria network from Larremore et al. (2013), the CSS from Krackhardt (1987), and the first synthetic network from Section 6.1. We discuss the data point marked "Gossip village 48" in further detail in Section 6.2.4 and Figure 5.

In this context, we echo the motivation of identifying layer interdependence in such social multilayer networks: that finding layer redundancies could justify a less extensive data-collection of future social networks in similar settings. We further explore this possibility with analysis in the next section.

6.2.4 Collection of Village Social Support Networks

We now turn our attention to two sets of multilayer social networks representing different types of interaction in 113 villages: 43 *microfinance villages* from Banerjee et al. (2013) and 70 *gossip villages* from Banerjee et al. (2019). The intent in studying these large collections of village networks is to see if the observation from Section 6.2.3, that Village 0 is layer redundant at level $\alpha = 0.05$, is common across multiple different networks of the same type. The survey questions defining each layer for these networks are different for each data source (see Appendix H): the 12 types of social support defining the layers in the networks of Banerjee et al. (2013) are different from the social support defining the 7 layers in the networks of Banerjee et al. (2019).

For each of these 113 villages we fix $K = 4$ and $K = 10$ and perform cross-validation under the independent link prediction task for both the layer independent and layer redundant NNTuck. We plot the test-AUCs of this multi-village link prediction task in Figure 9, where we also plot the test-AUC of the corresponding models in the Krackhardt, Village 0, and first synthetic network. Surprisingly, we note that the test-AUC for the layer redundant NNTuck is nearly equivalent to the test-AUC for the layer independent NNTuck for almost all of the village networks. We highlight the village network with the biggest difference between the two test-AUCs, labeled "Gossip village 48", and estimate a layer dependent

NNTuck with $K = 4$ and $C = 2$. With this model choice, we find that the network is layer dependent at level $\alpha = 0.05$ and interpret the corresponding $\hat{\boldsymbol{Y}}$ in Figure 5.

Finding evidence of layer redundancy in each of these 113 different village multilayer networks provides more evidence and motivation for using the NNTuck as a tool for survey design. Specifically, we see evidence that each of these different types of social affiliation and support (between 7 to 12 depending on the network, see Appendix H for information on the different types of relationships in each network) are all well modeled by the same generative process. If, in a future study, the same evidence was found in initial data collection, then before scaling the study to more villages a less expensive survey could ask less questions of the participants. However, even if all layers of data are collected, knowing that they are redundant could help enhance other structural properties in the network (e.g., Nayar et al., 2015; Taylor et al., 2016, 2017).

7. Conclusion

In this work we use the nonnegative Tucker decomposition (NNTuck) with KL-divergence as an extension of the stochastic block model (SBM) to multilayer networks. The NNTuck allows for layers in the network to have latent structure, just as the SBM allows for latent structure in the nodes of a single layer network. Using algebraic examples we show that the third factor matrix of the NNTuck both captures and incorporates information about layer interdependence in multilayer networks. We show that the multiplicative updates for minimizing the KL-divergence of the NNTuck are step-by-step equivalent to maximizing the log-likelihood of observing the network under the NNTuck model using expectation maximization. This equivalence generalizes a previously known result about matrices and motivates the use of this algorithm in the context of the NNTuck.

To use the NNTuck to validate layer dependence in empirical multilayer networks, we define three likelihood ratio tests (LRTs) to test layer independence, layer redundance, and layer dependence. Furthermore, we propose three methods for interpreting the third factor matrix of an NNTuck estimated for an empirical network. We propose cross-validation as a means for model selection and formalize two link prediction tasks for the multilayer setting. We use cross-validation, the LRTs, and the approaches for interpreting $\hat{\boldsymbol{Y}}$ to study a variety of synthetic and empirical multilayer networks. In doing so, we find that the Malaria multilayer network has independent layers, 113 different social support networks are layer redundant, and Krackhardt's cognitive social structure has layer dependence.

This work also lays the groundwork for diverse future work and applications. Given the observation in Section 6.2.4, that for many of the village multilayer networks we study the layers seem to be noisy observations from the same SBM, it would be interesting to explore how other models of network formation (e.g., the choice-based dynamic models in Overgoor et al. (2019)) uncover different characteristics amongst the layers that the SBM cannot identify. As discussed in Section 5, the difference in test-AUC between the layer independent NNTuck and the layer dependent NNTuck with $C = L$ is not fully understood and could be addressed in future work. Furthermore, some multilayer graphs (for instance, the `ogbl-wikikg2` multilayer network from Hu et al. (2020), which a reviewer brought to our attention) have such a structure where an edge in one layer determines the presence of an edge in another layer. It is unclear if the NNTuck would be well suited, or

justified, to be used to model such structureed multilayer networks, and inspecting this is an interesting subject for future work. Finally, an interesting future direction is understanding how the NNTuck can be made more interpretable under a Varimax rotation, following recent connections between Varimax and factor model inference in Rohe and Zeng (2020).

Understanding the layer dependencies in a multilayer network can inform the development of survey design, identify redundancies, or illuminate contextual connections. Moreover, the usefulness of finding latent structure in the layers motivates the use of latent-space models as a noise-free smoothing of the observed network, as proposed by Fisher and Pinter-Wollman (2021). As such, there is potential to use this work to understand layer dependence in a variety of applications where domain-specific knowledge can make use of the interpretations that the NNTuck provides.

Acknowledgments

IA acknowledges support from the NSF GRFP and the Knight-Hennessy Scholars Fellowship. DT acknowledges partial support from NSF (#DMS-2052720) and the Simons Foundation (#578333). JU acknowledges partial support ARO (#76582-NS-MUR) and NSF (#2143176). We would also like to thank our reviewers for insightful suggestions and Caterina De Bacco, Eleanor Power, Dan Larremore, and Samir Khan for helpful conversations.

Appendix A. Implementation Details

Tools for this work and an example jupyter notebook (all in python) can be found at `https://github.com/izabelaguiar/NNTuck`

The implementation of the multiplicative updates algorithm from Kim and Choi (2007), Algorithm 1, is done in python using the `tensorly` backend. The `tensorly` package (Kossaifi et al., 2016) already has an implementation of the multiplicative updates algorithm for estimating the nonnegative Tucker decomposition, although the implementation is for minimizing the least-squares loss between the tensor and its decomposition. We thus altered their implementation to include the multiplicative updates for minimizing KL-divergence. Our implementation uses dense matrix computations, and as such, does not take advantage of the sparse structure present in multilayer networks. Although the `tensorly` package does have a sparse option, their documentation describes that only *memory usage* is improved by using the sparse backend, and that using the sparse backend for decompositions actually takes much longer.

To present a scope of how computation time scales with network size, we plot the average time Algorithm 1 takes to converge across 20 random initializations for estimating both the layer redundant and layer independent NNTuck of synthetic multilayer networks with number of nodes varying from 50 to $10,000$ and number of layers varying from 5 to 20. As we see in Figure 10, the average time to convergence for the smallest network ($N = 50$, $L = 5$) was 0.125 seconds, and for the largest network ($N = 10,000$, $L = 20$) the average convergence time was 8.66 hours. As an upper limit for the computation we were able to do before running into memory limits, we generated a synthetic multilayer network with $N = 25,000$ nodes and $L = 5$ layers. Over 20 random initializations of Algorithm 1, estimating the layer redundant NNTuck and the layer independent NNTuck of this network took 12.43 hours and 18.044 hours, on average, respectively.

Efficient algorithms for estimating the nonnegative Tucker decomposition, such as those discussed in Zhou et al. (2015), minimize least-squares loss. Future work in extending the NNTuck for use in larger networks will necessitate a faster implementation of the algorithm discussed and used in this work, if not a completely different algorithm for minimizing KL-divergence.

Another implementation detail, discussed in Section 6.2, is that we select the NNTuck with the highest log-likelihood across 20 random initializations. In Fig. 11, we show the variation in maximal log likelihood as a function of number of random restarts and show that 20 random initializations is sufficient for estimating the NNTuck.

Appendix B. EM and NNTuck multiplicative updates equivalence

In this section we show the equivalence between the expectation maximization (EM) updates and the multiplicative updates for the nonnegative Tucker decomposition under a KL-divergence loss.

Recall Theorem 6, restated below,

Proposition 10 *Determining factor matrices $\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{Y}$ and the core tensor $\boldsymbol{\mathcal{G}}$ in the NNTuck by maximizing the log-likelihood using expectation maximization (EM) Eq.* (12) *is equivalent to using the multiplicative updates Eq.* (14) *to minimize KL-divergence.*

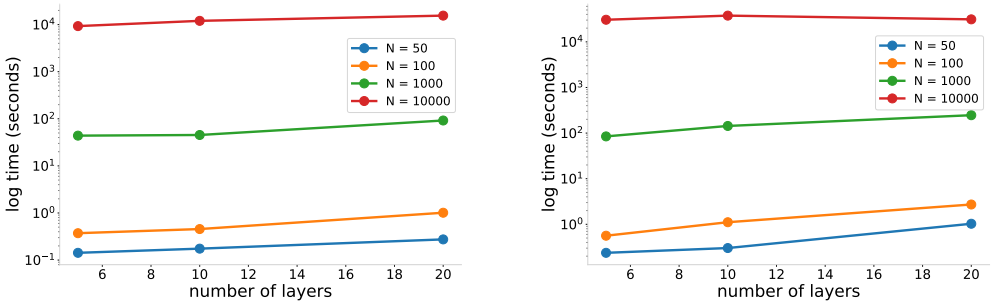


Figure 10: The time to convergence for estimating the layer redundant (left) and layer independent (right) NNTuck of synthetic networks of varying sizes, averaged across 20 random initializations, plotted on a log scale.

Consider using expectation maximization (EM) to reach a local maximum of the log-likelihood of observing $\boldsymbol{\mathcal{A}}$ under the model given by (8),

$$\mathcal{L}(\boldsymbol{\mathcal{A}}|\boldsymbol{U},\boldsymbol{V},\boldsymbol{Y},\boldsymbol{\mathcal{G}}) = \sum_{i,j,\alpha}\left[a_{ij\alpha}\log\sum_{k,\ell,\rho}u_{ik}v_{j\ell}y_{\alpha\rho}g_{k\ell\rho} - \sum_{k,\ell,\rho}u_{ik}v_{j\ell}y_{\alpha\rho}g_{k\ell\rho}\right], \tag{11}$$

in which case the following update equations are used

$$\begin{aligned} u_{ik} &= \frac{\sum_{j,\alpha}A_{ij\alpha}\sum_{\ell}\rho^{(\alpha)}_{ijk\ell}}{\sum_{\ell}\left(\sum_{j}v_{j\ell}\right)\left(\sum_{\alpha}g_{k\ell\alpha}\right)}, \\ v_{j\ell} &= \frac{\sum_{i,\alpha}A_{ij\alpha}\sum_{k}\rho^{(\alpha)}_{ijk\ell}}{\sum_{k}\left(\sum_{i}u_{ik}\right)\left(\sum_{\alpha}g_{k\ell\alpha}\right)}, \\ g_{k\ell\alpha} &= \frac{\sum_{ij}A_{ij\alpha}\rho^{(\alpha)}_{ijk\ell}}{\left(\sum_{i}u_{ik}\right)\left(\sum_{j}v_{j\ell}\right)}, \end{aligned} \tag{12}$$

where

$$\rho^{(\alpha)}_{ijk\ell} = \frac{u_{ik}v_{j\ell}y_{\alpha c}g_{k\ell c}}{\sum_{k'\ell'c'}u_{ik'}v_{j\ell'}y_{\alpha c'}g_{k'\ell'c'}}. \tag{13}$$

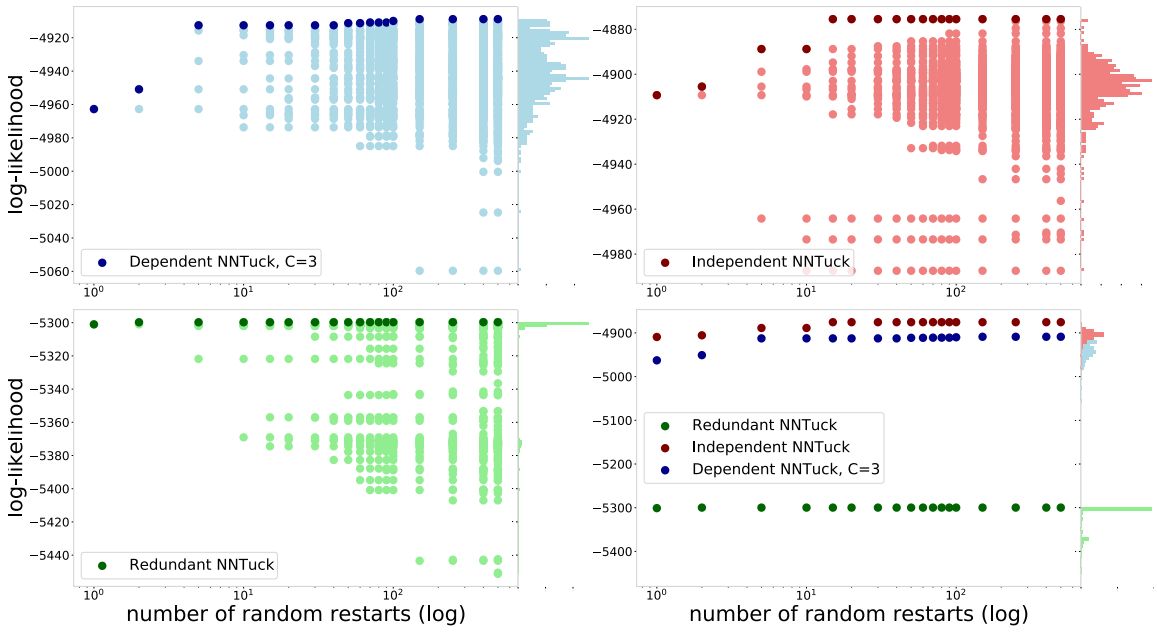


Figure 11: We show the log-likelihood of the dependent, independent, and dependent NNTucks for the Krackhardt multilayer network estimated using Algorithm 1. For each estimation we vary the number of random initializations on the horizontal axis, and for each we plot the maximal log-likelihood over that set with a bold point. In the last plot we plot the maximal log-likelihood for each NNTuck together. The histogram on the right hand side of each plot shows the distribution of the log-likelihood across all 500 random initializations. We see that the maximal log-likelihood does not vary greatly when using more than 20 random initializations, and thus determine that using 20 random initializations to estimate the NNTuck is appropriate.

Conversely, the multiplicative updates for nonnegative Tucker decomposition (NNTuck) under the KL-divergence loss as given by Kim and Choi (2007) are,

$$\begin{aligned}
\boldsymbol{U} &\leftarrow \boldsymbol{U} \circledast \frac{\left[\boldsymbol{A}_{(1)} / \left(\boldsymbol{U}\boldsymbol{G}_{\boldsymbol{U}}^{(1)}\right)\right] \boldsymbol{G}_{\boldsymbol{U}}^{(1)\top}}{\mathbf{1}\mathbf{z}_{\boldsymbol{U}}^{\top}}, \\
\boldsymbol{V} &\leftarrow \boldsymbol{V} \circledast \frac{\left[\boldsymbol{A}_{(2)} / \left(\boldsymbol{V}\boldsymbol{G}_{\boldsymbol{V}}^{(2)}\right)\right] \boldsymbol{G}_{\boldsymbol{V}}^{(2)\top}}{\mathbf{1}\mathbf{z}_{\boldsymbol{V}}^{\top}}, \\
\boldsymbol{Y} &\leftarrow \boldsymbol{Y} \circledast \frac{\left[\boldsymbol{A}_{(3)} / \left(\boldsymbol{Y}\boldsymbol{G}_{\boldsymbol{Y}}^{(3)}\right)\right] \boldsymbol{G}_{\boldsymbol{Y}}^{(3)\top}}{\mathbf{1}\mathbf{z}_{\boldsymbol{Y}}^{\top}}, \\
\boldsymbol{\mathcal{G}} &\leftarrow \boldsymbol{\mathcal{G}} \circledast \frac{(\boldsymbol{\mathcal{A}} / \hat{\boldsymbol{\mathcal{A}}}) \times_1 \boldsymbol{U}^{\top} \times_2 \boldsymbol{V}^{\top} \times_3 \boldsymbol{Y}^{\top}}{\boldsymbol{\mathcal{E}} \times_1 \boldsymbol{U}^{\top} \times_2 \boldsymbol{V}^{\top} \times_3 \boldsymbol{Y}^{\top}}, \\
z_{\boldsymbol{U}i} &= \sum_j (\boldsymbol{G}_{\boldsymbol{U}}^{(1)})_{ij}, \\
z_{\boldsymbol{V}i} &= \sum_j (\boldsymbol{G}_{\boldsymbol{V}}^{(2)})_{ij}, \\
z_{\boldsymbol{Y}i} &= \sum_j (\boldsymbol{G}_{\boldsymbol{Y}}^{(3)})_{ij}.
\end{aligned} \tag{14}$$

Above, $\boldsymbol{\mathcal{E}}$ is the all ones tensor of the same dimension as $\boldsymbol{\mathcal{A}}$, $\circledast$ and / denote elementwise multiplication and division, respectively, and the subscript $\cdot_{(\ell)}$ denotes the tensor ℓ-unfolding. $\boldsymbol{G}_{\boldsymbol{U}}^{(1)}$, $\boldsymbol{G}_{\boldsymbol{V}}^{(2)}$, and $\boldsymbol{G}_{\boldsymbol{Y}}^{(3)}$ are defined as

$$\boldsymbol{G}_{\boldsymbol{U}}^{(1)} = \left[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}\right]_{(1)}, \boldsymbol{G}_{\boldsymbol{V}}^{(2)} = \left[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}\right]_{(2)}, \boldsymbol{G}_{\boldsymbol{Y}}^{(3)} = \left[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}\right]_{(3)}.$$

We will make use of *tensor unfoldings* and the *tensor n-mode product* in the following equivalence proofs. To help guide intuition on how tensor unfoldings are used, we provide Figure 12 as one visualization of the three unfoldings of a third-order tensor $\boldsymbol{\mathcal{X}}$.

Equivalence of core tensor updates We first show that the updates to $\boldsymbol{\mathcal{G}}$ in Eq. (12) are equivalent to the updates to $\boldsymbol{\mathcal{G}}$ in Eq. (14).

Proof The (i, j, α) entry of $\hat{\boldsymbol{\mathcal{A}}} := \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}$ is

$$\hat{A}_{ij\alpha} = \sum_{k'\ell'p'} u_{ik'} v_{j\ell'} y_{\alpha p'} g_{k'\ell'p'},$$

and therefore the update to $g_{k\ell\alpha}$ can be rewritten as,

$$\begin{aligned}
g_{k\ell p} &= \frac{\sum_{ij\alpha} \left(A_{ij\alpha} / \hat{A}_{ij\alpha}\right) u_{ik} v_{j\ell} y_{\alpha p} g_{k\ell p}}{\left(\sum_i u_{ik}\right) \left(\sum_j v_{j\ell}\right) \left(\sum_\alpha y_{\alpha p}\right)} \\
&= g_{k\ell p} \cdot \frac{\sum_{ij\alpha} \left(A_{ij\alpha} / \hat{A}_{ij\alpha}\right) u_{ik} v_{j\ell} y_{\alpha p}}{\left(\sum_i u_{ik}\right) \left(\sum_j v_{j\ell}\right) \left(\sum_\alpha y_{\alpha p}\right)}.
\end{aligned}$$

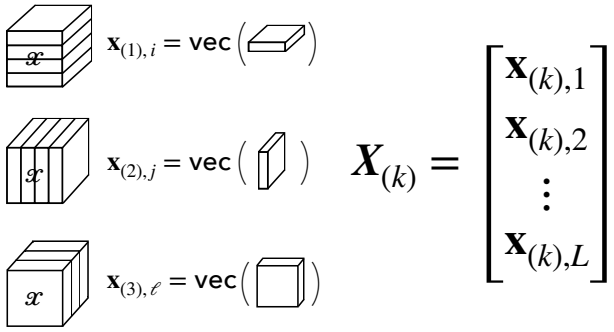


Figure 12: The k-unfolding of a tensor can be described by vertically stacking vectorizations of one of its slices. Above, in descending order, we see the horizontal, lateral, and frontal slices of a tensor. Vectorizing each slice and vertically stacking them gives the 1-, 2-, and 3-unfolding of tensor $\boldsymbol{\mathcal{X}}$, respectively. For $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N\times M\times L}$, then the dimensions of the three unfoldings are $\boldsymbol{X}_{(1)} \in \mathbb{R}^{M\times NL}$, $\boldsymbol{X}_{(2)} \in \mathbb{R}^{N\times ML}$, and $\boldsymbol{X}_{(3)} \in \mathbb{R}^{L\times NM}$.

Note the (a, b, c) element of the tensor mode-1 product of the following,

$$\left[\left(\frac{\boldsymbol{\mathcal{A}}}{\hat{\boldsymbol{\mathcal{A}}}}\right) \times_1 \boldsymbol{U}^\top\right]_{abc} = \sum_i \left(\frac{\boldsymbol{\mathcal{A}}}{\hat{\boldsymbol{\mathcal{A}}}}\right)_{ibc} u_{ia}. \tag{15}$$

Then,

$$\begin{aligned} \left\{\left[\left(\frac{\boldsymbol{\mathcal{A}}}{\hat{\boldsymbol{\mathcal{A}}}}\right) \times_1 \boldsymbol{U}^\top\right] \times_2 \boldsymbol{V}^\top \times_3 \boldsymbol{Y}^\top\right\}_{k\ell p} &= \sum_\alpha \left[\left(\frac{\boldsymbol{\mathcal{A}}}{\hat{\boldsymbol{\mathcal{A}}}}\right) \times_1 \boldsymbol{U}^\top \times_2 \boldsymbol{V}^\top\right]_{k\ell p} y_{\alpha p} \\ &= \sum_{j\alpha} \left[\left(\frac{\boldsymbol{\mathcal{A}}}{\hat{\boldsymbol{\mathcal{A}}}}\right) \times_1 \boldsymbol{U}^\top\right]_{kj\alpha} v_{j\ell} y_{\alpha p} \\ &= \sum_{j\alpha} \left(\sum_i \left(\frac{\boldsymbol{\mathcal{A}}}{\hat{\boldsymbol{\mathcal{A}}}}\right)_{ij\alpha} u_{ik}\right) v_{j\ell} y_{\alpha p} \\ &= \sum_{ij\alpha} \left(A_{ij\alpha}/\hat{A}_{ij\alpha}\right) u_{ik} v_{j\ell} y_{\alpha p}. \end{aligned} \tag{16}$$

Now, note the (a, b, c) element of the tensor mode-1 product of the following,

$$\left[\boldsymbol{\mathcal{E}} \times_1 \boldsymbol{U}^\top\right]_{abc} = \sum_i E_{ibc} u_{ia} = \sum_i u_{ia}. \tag{17}$$

Then,

$$\begin{aligned}\left\{\left[\boldsymbol{\mathcal{E}} \times_1 \boldsymbol{U}^\top\right] \times_2 \boldsymbol{V}^\top \times_3 \boldsymbol{Y}^\top\right\}_{k\ell p} &= \sum_{\alpha} \left[\boldsymbol{\mathcal{E}} \times_1 \boldsymbol{U}^\top \times_2 \boldsymbol{V}^\top\right]_{k\ell\alpha} y_{\alpha p} \\ &= \sum_{j\alpha} \left[\boldsymbol{\mathcal{E}} \times_1 \boldsymbol{U}^\top\right]_{kj\alpha} v_{j\ell} y_{\alpha p} \\ &= \sum_{j\alpha} \left(\sum_i u_{ik}\right) v_{j\ell} y_{\alpha p} \\ &= \left(\sum_i u_{ik}\right)\left(\sum_j v_{j\ell}\right)\left(\sum_\alpha y_{\alpha p}\right).\end{aligned} \tag{18}$$

Therefore, focusing on the NNTuck multiplicative update of core tensor $\boldsymbol{\mathcal{G}}$, we see that the update to the (k, ℓ, α) element of the core tensor is,

$$\begin{aligned} g_{k\ell p} &\leftarrow g_{k\ell p} \cdot \frac{[(\boldsymbol{\mathcal{A}}/\hat{\boldsymbol{\mathcal{A}}}) \times_1 \boldsymbol{U}^\top \times_2 \boldsymbol{V}^\top]_{k\ell\alpha}}{[\boldsymbol{\mathcal{E}} \times_1 \boldsymbol{U}^\top \times_2 \boldsymbol{V}^\top \times_3 \boldsymbol{Y}^\top]_{k\ell p}} \quad \Leftrightarrow \\ g_{k\ell p} &\leftarrow g_{k\ell p} \cdot \frac{\sum_{ij\alpha}\left(A_{ij\alpha}/\hat{A}_{ij\alpha}\right) u_{ik} v_{j\ell} y_{\alpha p}}{\left(\sum_i u_{ik}\right)\left(\sum_j v_{j\ell}\right)\left(\sum_\alpha y_{\alpha p}\right)}, \end{aligned} \tag{19}$$

which is equivalent to the update to the (k, ℓ, α) element of $\boldsymbol{\mathcal{G}}$ in Eq. (12). ■

Equivalence of factor matrix $\boldsymbol{U}$ updates For showing the equivalence in updates to factor matrix $\boldsymbol{U}$, the following identity is used multiple times. For tensor $\mathcal{X}$,

$$\sum_{j\alpha} X_{ij\alpha} = \sum_j \boldsymbol{X}_{(1)ij}, \tag{20}$$

where $\boldsymbol{X}_{(1)}$ denotes the 1-unfolding of $\boldsymbol{\mathcal{X}}$.

Proof Consider the EM update to u_{ik} from 12. Again,

$$\hat{A}_{ij\alpha} = \sum_{k'\ell'p'} u_{ik'} v_{j\ell'} y_{\alpha p'} g_{k'\ell'p'},$$

and so the EM update to u_{ik} can be rewritten as,

$$\begin{aligned} u_{ik} &= \frac{\sum_{j\alpha}\left(A_{ij\alpha}/\hat{A}_{ij\alpha}\right)\sum_{\ell p} u_{ik} v_{j\ell} y_{\alpha p} g_{k\ell p}}{\sum_{\ell p}\left(\sum_j v_{j\ell}\right)\left(\sum_\alpha y_{\alpha p} g_{k\ell p}\right)} \\ &= u_{ik} \cdot \frac{\sum_{j\alpha}\left(A_{ij\alpha}/\hat{A}_{ij\alpha}\right)\sum_{\ell p} v_{j\ell} y_{\alpha p} g_{k\ell p}}{\sum_{\ell p}\left(\sum_j v_{j\ell}\right)\left(\sum_\alpha y_{\alpha p} g_{k\ell p}\right)}. \end{aligned} \tag{21}$$

Now note that from the above equation we can identify out

$$\sum_{\ell p} v_{j\ell} y_{\alpha p} g_{k\ell p} = [\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{kj\alpha}. \tag{22}$$

For vector $\mathbf{z}$ such that $z_i = \sum_j [\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)ij}$, note that

$$\begin{aligned}\left[\mathbf{1}\mathbf{z}^\top\right]_{ik} = (1)(z_k) = z_k &= [\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V}\times)3]_{(1)kj} \\ &= \sum_{j\alpha} [\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{kj\alpha} \\ &= \sum_{j\alpha}\sum_{\ell p} g_{k\ell p} v_{j\ell} y_{\alpha p} \\ &= \sum_{\ell p}\sum_{j\alpha} g_{k\ell p} y_{\alpha p} v_{j\ell} \\ &= \sum_{\ell p}\left(\sum_j v_{j\ell}\right)\left(\sum_\alpha y_{\alpha p} g_{k\ell\alpha}\right).\end{aligned} \tag{23}$$

Substituting (22) and (23) into Eq. (21), we have that

$$\begin{aligned}u_{ik} &= u_{ik} \cdot \frac{\sum_{j\alpha}\left(A_{ij\alpha}/\hat{A}_{ij\alpha}\right)[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{kj\alpha}}{[\mathbf{1}\mathbf{z}^\top]_{ik}} \\ &= u_{ik} \cdot \frac{\sum_{j\alpha}\left(A_{ij\alpha}/\hat{A}_{ij\alpha}\right)[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)kj}}{[\mathbf{1}\mathbf{z}^\top]_{ik}} \\ &= u_{ik} \cdot \frac{\sum_{j\alpha}\left(A_{ij\alpha}/\hat{A}_{ij\alpha}\right)[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]^\top_{(1)jk}}{[\mathbf{1}\mathbf{z}^\top]_{ik}}.\end{aligned} \tag{24}$$

Then using Eq. (20) we have that,

$$\begin{aligned}&\sum_{j\alpha} A_{ij\alpha} = \sum_j \boldsymbol{A}_{(1)ij}, \\ &\sum_{j\alpha} \hat{A}_{ij\alpha} = \sum_j \hat{\boldsymbol{A}}_{(1)ij} = \sum_j (\boldsymbol{U}[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)})_{ij}, \\ &\sum_{j\alpha} [\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{kj\alpha} = \sum_j [\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)kj} = \sum_j [\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]^\top_{(1)jk}.\end{aligned} \tag{25}$$

In the second line of Eq. (25) above, we use that

$$\begin{aligned}\hat{\boldsymbol{\mathcal{A}}} &= \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y} \\ &= \boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y} \times_1 \boldsymbol{U} \\ &= (\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}) \times_1 \boldsymbol{U},\end{aligned}$$

and thus using the identity $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \boldsymbol{B} \Rightarrow \boldsymbol{Y}_{(n)} = \boldsymbol{B}\boldsymbol{X}_{(n)}$, we get $\hat{\boldsymbol{A}}_{(1)} = \boldsymbol{U}[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)}$.

Therefore,

$$\begin{aligned}u_{ik} &= u_{ik} \cdot \frac{\sum_j (\boldsymbol{A}_{(1)}/(\boldsymbol{U}[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)}))_{ij}[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]^\top_{(1)jk}}{[\mathbf{1}\mathbf{z}^\top]_{ik}} \\ &= u_{ik} \cdot \frac{\left[(\boldsymbol{A}_{(1)}/(\boldsymbol{U}[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)}))[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]^\top_{(1)}\right]_{ik}}{[\mathbf{1}\mathbf{z}^\top]_{ik}}.\end{aligned} \tag{26}$$

This is equivalent to the ik update of $\boldsymbol{U}$ given by Eq. (14). ■

Equivalence of factor matrix $\boldsymbol{V}$ updates In connecting the updates to factor matrix $\boldsymbol{V}$, the following identity is used multiple times. For tensor $\mathcal{X}$,

$$\sum_{i\alpha} X_{ij\alpha} = \sum_{i} \boldsymbol{X}_{(2)ji}, \tag{27}$$

where $\boldsymbol{X}_{(1)}$ denotes the (1)-unfolding of $\boldsymbol{\mathcal{X}}$.

Proof Consider the EM update to $v_{j\ell}$ given by Eq. (12). Because

$$\hat{A}_{ij\alpha} = \sum_{k'\ell'p'} u_{ik'} v_{j\ell'} y_{\alpha p'} g_{k'\ell'p'},$$

then the EM update to $v_{j\ell}$ can be rewritten as,

$$\begin{aligned} v_{j\ell} &= \frac{\sum_{i\alpha} \left(A_{ij\alpha} / \hat{A}_{ij\alpha} \right) \sum_{kp} u_{ik} v_{j\ell} y_{\alpha p} g_{k\ell p}}{\sum_{kp} \left(\sum_i u_{ik} \right) \left(\sum_\alpha y_{\alpha p} g_{k\ell p} \right)} \\ &= v_{j\ell} \cdot \frac{\sum_{i\alpha} \left(A_{ij\alpha} / \hat{A}_{ij\alpha} \right) \sum_{kp} u_{ik} y_{\alpha p} g_{k\ell p}}{\sum_{kp} \left(\sum_j u_{ik} \right) \left(\sum_\alpha y_{\alpha p} g_{k\ell p} \right)}. \end{aligned} \tag{28}$$

Now note that we can again identify out

$$\sum_{kp} u_{ik} y_{\alpha p} g_{k\ell p} = [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{i\ell\alpha}. \tag{29}$$

And again, for vector $\mathbf{z}$ such that $z_j = \sum_i [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{Y}]_{(2)ji}$, note that

$$\begin{aligned} \left[\mathbf{1}\mathbf{z}^\top \right]_{j\ell} = (1)(z_\ell) = z_\ell &= \sum_i [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{Y}]_{(2)\ell i} \\ &= \sum_{i\alpha} [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{i\ell\alpha} \\ &= \sum_{i\alpha} \sum_{kp} u_{ik} y_{\alpha p} g_{k\ell p} \\ &= \sum_{kp} \sum_{i\alpha} u_{ik} y_{\alpha p} g_{k\ell p} \\ &= \sum_{kp} \left(\sum_i u_{ik} \right) \left(\sum_\alpha y_{\alpha p} g_{k\ell p} \right). \end{aligned} \tag{30}$$

Substituting together (29) and (30) into the EM updates Eq. (28), we have that

$$v_{j\ell} = v_{j\ell} \cdot \frac{\sum_{i\alpha} \left(A_{ij\alpha} / \hat{A}_{ij\alpha} \right) [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{i\ell\alpha}}{[\mathbf{1}\mathbf{z}^\top]_{j\ell}}. \tag{31}$$

Then,

$$\begin{aligned} \sum_{i\alpha} A_{ij\alpha} &= \sum_i \boldsymbol{A}_{(2)ji}, \\ \sum_{i\alpha} \hat{A}_{ij\alpha} = \sum_i \hat{\boldsymbol{A}}_{(2)ji} &= \sum_i (\boldsymbol{V}[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)})_{ji}, \\ \sum_{i\alpha} [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{i\ell\alpha} = \sum_i [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)\ell i} &= \sum_i [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]^\top_{(2)i\ell}. \end{aligned} \tag{32}$$

In the second line of 32 above, we use that

$$\hat{\boldsymbol{\mathcal{A}}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y} \times_2 \boldsymbol{V} = (\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}) \times_2 \boldsymbol{V},$$

and thus using the identity $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \boldsymbol{B} \Rightarrow \boldsymbol{Y}_{(n)} = \boldsymbol{B}\boldsymbol{X}_{(n)}$, we get $\hat{\boldsymbol{A}}_{(2)} = \boldsymbol{V}[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)}$.

Therefore,

$$\begin{aligned} v_{j\ell} &= v_{j\ell} \cdot \frac{\sum_i (\boldsymbol{A}_{(2)} / (\boldsymbol{V}[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)}))_{ji} [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]^\top_{(2)i\ell}}{[\mathbf{1}\mathbf{z}^\top]_{j\ell}} \\ &= v_{j\ell} \cdot \frac{\left[(\boldsymbol{A}_{(2)} / (\boldsymbol{V}[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)})) [\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]^\top_{(2)} \right]_{j\ell}}{[\mathbf{1}\mathbf{z}^\top]_{j\ell}} \end{aligned} \tag{33}$$

This is equivalent to the $j\ell$ update of $\boldsymbol{V}$ given by Eq. (14). ■

Showing the equivalence for the updates to $\boldsymbol{Y}$ follow exactly as those above, and thus we do not reproduce it here. Therefore, we have shown that the EM updates given by Eq. (12) are step-by-step equivalent to the multiplicative updates given by Eq. (14).

Appendix C. Masked updates

In this section we discuss the algorithmic changes to the multiplicative updates from Kim and Choi (2007) for the nonnegative Tucker decomposition under a KL-divergence loss to allow for masking, as used during cross-validated model evaluation. This section describes the tensor completion problem wherein we wish to build the NNTuck using only observed entries. Consider that some of the entries of adjacency tensor $\boldsymbol{\mathcal{A}}$ are unobserved. We wish to build the NNTuck of $\boldsymbol{\mathcal{A}}$ using *only* the observed entries, but we want the reconstruction $\hat{\boldsymbol{\mathcal{A}}}$ to predict the unobserved entries. This *tensor completion* problem is how we train on 80% of the entries of $\hat{\boldsymbol{\mathcal{A}}}$ for the five-fold cross-validation in Section 5. Assume that there is a set $\mathcal{I}$ such that

$$\mathcal{I} := \{(i, j, \alpha) \mid A_{ij\alpha} \text{ is unobserved}\}. \tag{34}$$

We want to rederive the update rules from Kim and Choi (2007) for nonnegative Tucker decomposition to only account for the observed entries of $\boldsymbol{\mathcal{A}}$. To do so, we introduce a *masking tensor*, $\boldsymbol{\mathcal{M}}$ such that,

$$M_{ij\alpha} = \begin{cases} 0 & \text{if } (i, j, \alpha) \in \mathcal{I} \\ 1 & \text{if } (i, j, \alpha) \notin \mathcal{I}. \end{cases}$$

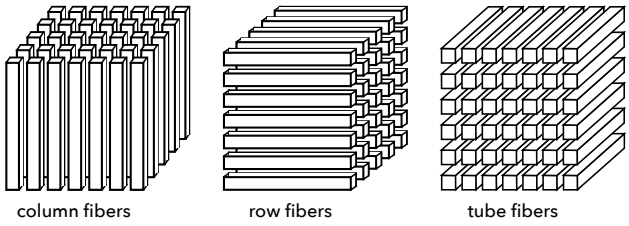


Figure 13: The row, column, and tube *fibers* of a third-order tensor. This figure has been adapted from Figure 2 in the tensor review by Kolda and Bader (2009). In the *tubular* link prediction task in Section 5, entire tubes of the adjacency tensor are missing i.i.d. with uniform probability.

Then re-deriving the update rules from the log-likelihood and EM approach, and after re-tensorizing them, the update rules for factor matrices $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{Y}$ and for core tensor $\boldsymbol{\mathcal{G}}$ are,

$$\boldsymbol{U} = \boldsymbol{U} \circ \frac{[\boldsymbol{M}_{(1)} \circ \boldsymbol{A}_{(1)} / \hat{\boldsymbol{A}}_{(1)}][\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)}^{\top}}{\boldsymbol{M}_{(1)}[\boldsymbol{\mathcal{G}} \times_2 \boldsymbol{V} \times_3 \boldsymbol{Y}]_{(1)}^{\top}}, \tag{35}$$

$$\boldsymbol{V} = \boldsymbol{V} \circ \frac{[\boldsymbol{M}_{(2)} \circ \boldsymbol{A}_{(2)} / \hat{\boldsymbol{A}}_{(2)}][\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)}^{\top}}{\boldsymbol{M}_{(2)}[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_3 \boldsymbol{Y}]_{(2)}^{\top}}, \tag{36}$$

$$\boldsymbol{Y} = \boldsymbol{Y} \circ \frac{[\boldsymbol{M}_{(3)} \circ \boldsymbol{A}_{(3)} / \hat{\boldsymbol{A}}_{(3)}][\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}]_{(3)}^{\top}}{\boldsymbol{M}_{(3)}[\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}]_{(3)}^{\top}}, \tag{37}$$

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{G}} \circ \frac{[\boldsymbol{\mathcal{M}} \circ \boldsymbol{\mathcal{A}} / \hat{\boldsymbol{\mathcal{A}}}] \times_1 \boldsymbol{U}^{\top} \times_2 \boldsymbol{V}^{\top} \times_3 \boldsymbol{Y}^{\top}}{\boldsymbol{\mathcal{M}} \times_1 \boldsymbol{U}^{\top} \times_2 \boldsymbol{V}^{\top} \times_3 \boldsymbol{Y}^{\top}}. \tag{38}$$

In all of the above, $\circ$ denotes elementwise multiplication and / denotes elementwise division.

Appendix D. Variation in latent structure parameter K

Here, we discuss in further detail the decisions in varying K as we do in the cross-validation tasks from main text. The parameter K defines the dimension of the latent structure in the node set of the network. As such, K can vary from 1 to N, however in the cross-validation tasks in the main text we only vary it from 2 to 20 (or from 2 to 12 for the Krackhardt 1987 dataset). In interpreting the ends of the possible range of K values, we note that $K = 1$ assumes that each node in the multilayer network belongs to the same latent space. Conversely, $K = N$ allows each node to belong to its own latent space (and, therefore, assumes that there is no latent structure in the node set of the network). To consider how test-AUC varies with K values beyond those which we considered in the main text, we here extend the range of K in two of our multilayer network datasets.

First, we consider how test-AUC varies for larger values of K in the Village 0 multilayer social network. In this dataset, there are 843 nodes, and therefore the largest possible value of K is 843. However, because it is computationally unreasonable to consider all parameter

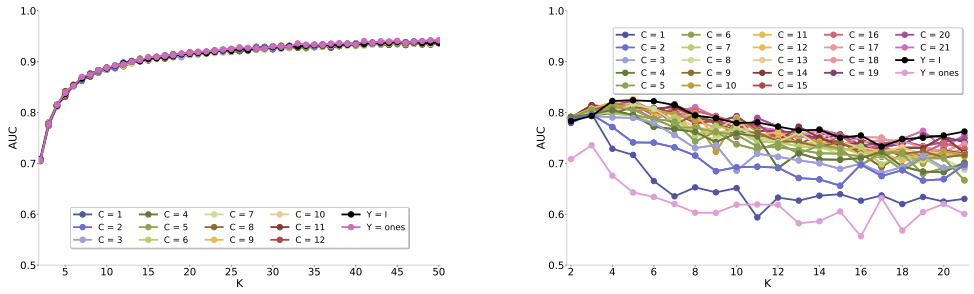


Figure 14: The test-AUC for the fivefold tubular cross-validation task for expanded range of K for (left) the Village 0 multilayer network from Banerjee et al. (2013) and (right) the cognitive social structure multilayer network from Krackhardt (1987).

combinations of K and C for such a large K, and because the model where K is this large lacks interpretability (in village networks like the one examined here, the latent node space tends to identify caste membership, see, e.g., De Bacco et al. (2017)), we increase our range of K to be from $K = 2$ to $K = 50$. We plot the test-AUC for the fivefold tubular cross-validation task for this expanded range of K in Figure 14 (left). We see that test-AUC improves with increasing K, but that this increase follows the pattern of increase we saw with the smaller range of K. More importantly, the larger K values do not differentiate the layer redundant NNTuck from the layer dependent or layer independent NNTuck model specifications: considering larger values of K does not change the conclusions we draw in the main text.

Next, we consider extending our range of K to its maximum of $K = N$ for the cognitive social structure multilayer network from Krackhardt (1987). Although letting $K = N$ assumes that there is no latent structure in the nodes of the network, we take the opportunity in this dataset with a relatively small node set to vary K to its maximum. We plot the test-AUC for the fivefold tubular cross-validation task in Figure 14 (right). Again, we see that test-AUC doesn't improve when considering larger values of K and, notably, our conclusions from the main text do not change with this increased variation.

Appendix E. Interpretability of the $\boldsymbol{Y}$ factor matrix

In this section we show the steps necessary to rewrite the core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{K\times K\times C}$ of the NNTuck in the basis of C unique reference layers, and how to find the corresponding $\hat{\boldsymbol{Y}}^*$ matrix, as discussed in Section 4.2. Take as given a set of C unique reference layers, denoted $r^* = \{r_1, \ldots, r_C\}$ where $r_i \in \{1, \ldots, L\}$. Define $\boldsymbol{\mathcal{G}}^*$ whose frontal slices are $\boldsymbol{\mathcal{G}}_\ell^* = \sum_{i=1}^{C} y_{r_\ell, i}\boldsymbol{G}_\ell$ for $\ell = 1, \ldots, C$. Define matrix $\boldsymbol{Y}^*$ such that rows r^* of $\boldsymbol{Y}^*$ are identity rows. Specifically, for row ℓ of the matrix,

$$\mathbf{y}_\ell^* = \begin{cases} e_\ell & \text{if } \ell \in r^*, \\ \mathbf{y}_\ell^* & \text{otherwise.} \end{cases}$$

Define $\bar{r} := \{\ell \mid \ell \notin r^*\}$. Then row $\mathbf{y}^*_\ell$ for $\ell \in \bar{r}$ satisfies,

$$y_{\ell,1}\boldsymbol{G}_1 + y_{\ell,2}\boldsymbol{G}_2 + \cdots + y_{\ell,C}\boldsymbol{G}_C = y^*_{\ell,1}\left(\sum_{i=1}^{C} y_{r_1,i}\boldsymbol{G}_i\right) + y^*_{\ell,2}\left(\sum_{i=1}^{C} y_{r_2,i}\boldsymbol{G}_i\right) + \cdots + y^*_{\ell,C}\left(\sum_{i=1}^{C} y_{r_C,i}\boldsymbol{G}_i\right).$$

This gives us the system of equations,

$$\begin{aligned} y_{\ell,1} &= y^*_{\ell 1}y_{r_1,1} + y^*_{\ell,2}y_{r_2,1} + \cdots y^*_{\ell,C}y_{r_C,1} \\ y_{\ell,2} &= y^*_{\ell 2}y_{r_1,2} + y^*_{\ell,2}y_{r_2,2} + \cdots y^*_{\ell,C}y_{r_C,2} \\ &\vdots \\ y_{\ell,C} &= y^*_{\ell C}y_{r_1,C} + y^*_{\ell,2}y_{r_2,C} + \cdots y^*_{\ell,C}y_{r_C,C}. \end{aligned}$$

Note that the first equation is the inner product between the first column of $\boldsymbol{Y}$ subsetted to the rows in r^* with the unknown vector $\mathbf{y}^*$. That is,

$$y_{\ell,1} = \mathbf{y}^\top_{r^*,1}\mathbf{y}^{*\top}_\ell.$$

Then let $\boldsymbol{Y}_{r^*}$ be the matrix $\boldsymbol{Y}$ subsetted to the rows in r^*. Then the ℓth row of matrix $\boldsymbol{Y}^*$, denoted $\mathbf{y}^*_\ell$, satisfies the linear system

$$\mathbf{Y}^\top_{r^*}\mathbf{y}^{*\top}_\ell = \mathbf{y}^\top_\ell. \tag{39}$$

Because there are $(L - C)$ unknown rows of matrix $\boldsymbol{Y}^*$, there will be $(L - C)$ such linear systems for each $\ell \in \bar{r}$.

Let $\boldsymbol{Y}^*_{\bar{r}}$ be the matrix $\boldsymbol{Y}^*$ subsetted to the rows *not* in r^*. Then,

$$\boldsymbol{Y}^\top_{r^*}\boldsymbol{Y}^{*\top}_{\bar{r}} = \boldsymbol{Y}^T_{\bar{r}} \iff \boldsymbol{Y}_{\bar{r}} = \boldsymbol{Y}^*_{\bar{r}}\boldsymbol{Y}_{r^*}. \tag{40}$$

Note that $\boldsymbol{Y}_{r^*} \in \mathbb{R}^{C\times C}$ and is invertible if and only if the rows of $\boldsymbol{Y}$ defined by r^* are not linearly dependent. This highlights the importance in choosing the "correct" reference rows. Even though in practice it's unlikely that any two rows, even if poorly chosen, will be exactly linearly dependent, if they are close then $\boldsymbol{Y}_{r^*}$ will not be invertible and the transformed $\boldsymbol{Y}^*$ matrix will not be interpretable.

The best reference layers are often determined by domain knowledge. Absent a principled approach but seeking to make the matrix interpretable, we propose the following heuristic for choosing the best reference layers: choose the C layers such that $\det(\boldsymbol{Y}_{r^*})$ is furthest from zero. Although searching over the entire space amounts to finding the determinant of $\binom{L}{C}$ different submatrices and isn't practical, one can instead compare the determinant of a handful of different submatrices, where reference layers can be chosen with a combination of (perhaps weak) domain expertise and by inspection of the $\boldsymbol{Y}$ matrix.

Appendix F. Likelihood ratio test with varying network size

In this section we explore the relationship between the power of the standard likelihood ratio test and the number of samples—in our context, the number of nodes and layers in the network. As discussed in Section 4.1, the analysis of Wilks' theorem, which provides the foundation for the LRT, is asymptotic. Thus, it is important to explore how the size of the multilayer networks we study impact the power of the statistical tests we use to evaluate layer interdependence.

To explore this, we conduct the following numerical experiment. We generate multiple layer redundant multilayer networks where we vary the number of nodes from 50 to 1000 and number of layers from 2 to 20. Each layer of the multilayer network is drawn from the same $K = 2$ SBM, where the first half of the nodes belong to the first latent node space and the second half of the nodes belong to the second latent node space. We then estimate a layer redundant NNTuck and a $C = 2$ layer dependent NNTuck, each by using the estimation with the highest log-likelihood over 20 random initializations. We use the log-likelihoods from each estimation to conduct a layer redundance LRT. We plot the p-value from each test in Figure 15.

We see that we (correctly) fail to reject H_0 for all of the synthetic networks of varying L and N. The p-value corresponding to the LRT is consistently far above $\alpha = 0.05$, with no discernible pattern corresponding to either the number of nodes N or the number of layers L. Although the concept of a p-value is not the same for the split-LRT, conducting the same experiments using the split-LRT we also fail to reject H_0 for all of the synthetic networks.

Appendix G. Likelihood ratio test without regularity conditions

In this section we briefly discuss the split-likelihood ratio test from Wasserman et al. (2020) as it relates to the LRT for the NNTuck. The split-LRT requires no regularity conditions, and is thus appealing in the setting at hand, wherein Wilks' theorem is not satisfied: the NNTuck is both non-identifiable and has a non-convex log-likelihood.

To describe the split-LRT we first define the following notation. The sets D_0 and D_1 split the data into two sets, each roughly equal in size, represented by masking tensors $\boldsymbol{\mathcal{M}}_0 \in \{0,1\}^{N\times N\times L}$ and $\boldsymbol{\mathcal{M}}_1 \in \{0,1\}^{N\times N\times L}$ where

$$\boldsymbol{\mathcal{M}}_0(i,j,k) = \begin{cases} 1 & \text{if } (i,j,k) \in D_0, \\ 0 & \text{if } (i,j,k) \in D_1 \end{cases} \text{ and } \boldsymbol{\mathcal{M}}_1(i,j,k) = \begin{cases} 1 & \text{if } (i,j,k) \in D_1, \\ 0 & \text{if } (i,j,k) \in D_0 \end{cases}.$$

The hypotheses for the split-LRT are

H_0: The network comes from the nested (layer redundant or dependent) NNTuck,

H_1: The network comes from the full (layer independent) NNTuck.

Parameter $\hat{\theta}_1 = [\hat{\boldsymbol{\mathcal{G}}}_1, \hat{\boldsymbol{U}}_1, \hat{\boldsymbol{V}}_1, \boldsymbol{I}]$ is *any* estimator under the layer independent NNTuck estimated only on D_1 using masking tensor $\boldsymbol{\mathcal{M}}_1$. Parameter $\hat{\theta}_0 = [\hat{\boldsymbol{\mathcal{G}}}_0, \hat{\boldsymbol{U}}_0, \hat{\boldsymbol{V}}_0, \hat{\boldsymbol{Y}}_0]$ is the *maximum likelihood estimator* under the nested NNTuck estimated only on D_0 using mask-

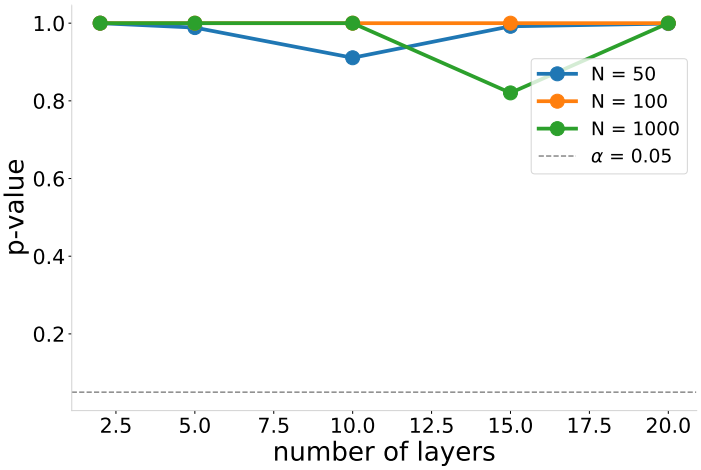


Figure 15: The p-values corresponding to the layer redundance LRT on synthetic, layer redundant multilayer networks of varying size, both in number of nodes and number of layers. The dashed grey line corresponds to a p-value of 0.05, at which we would reject the null (layer redundant) hypothesis with significance at 0.05. For every p-value above the dashed grey line, we would fail to reject the null hypothesis. Recall that the layer redundance LRT compares the layer redundant NNTuck to the layer dependent NNTuck with $C = 2$. In this plot we see that the LRT (correctly) fails to reject the null hypothesis for all synthetic networks of varying L and N.

ing tensor $\boldsymbol{\mathcal{M}}_0$. The null hypothesis is rejected at significance level α if

$$\log \frac{\mathcal{L}_0(\hat{\theta}_1)}{\mathcal{L}_0(\hat{\theta}_0)} > \log 1/\alpha, \tag{41}$$

where

$$\mathcal{L}(\theta) = \prod_{(i,j,k)\in D_0} p_\theta(\boldsymbol{\mathcal{A}}_{ijk}) \tag{42}$$

is the likelihood associated with θ only measured over the set D_0. Then, under no regularity conditions, the probability that we falsely reject the null hypothesis is,

$$\Pr_{H_0}(reject) \leq \alpha. \tag{43}$$

In applying the split-LRT to testing for layer redundance, dependence, and independence in multilayer networks, we use the same tests as discussed in Section 4 but using the split-LRT defined in Eq. (41) to reject or fail to reject the null hypothesis. Our results from both the standard and split-LRT are in Table 3.

We estimate $\hat{\theta}_1$ with the layer independent NNTuck with the highest log-likelihood over 20 random initializations of Algorithm 1 (see Appendix A). We estimate $\hat{\theta}_0$ with the redundant or dependent NNTuck with the highest log-likelihood over 50 random initializations of Algorithm 1. We increase the number of random initializations for estimating the nested NNTuck because the analysis in Wasserman et al. (2020) requires $\hat{\theta}_0$ to be the maximum likelihood estimator (MLE), something we cannot guarantee due to non-convexity. A more suitable approach to adapting the split-LRT to this setting would be to instead find $\hat{\theta}_0$ corresponding to the maximum of a (convex) proper relaxation to the likelihood of the NNTuck. An interesting topic of future work is finding a multilayer extension of the semidefinite relaxation of the MLE for the (single layer) stochastic block model developed by Amini and Levina (2018).

Appendix H. Questions generating the social multilayer networks

We reproduce the questions generating the multilayer social support networks from the Banerjee et al. (2013) and Banerjee et al. (2019). For specific information about the context, original findings, or survey instruments of the research, please refer to the original papers.

Microfinance Villages (Banerjee et al., 2013) The 43 multilayer networks from this research, representing different villages in Karnataka, India, were the result of asking individuals about 12 different types of support. Each layer in the resulting network corresponds to the following relationships.

```
1: Those from whom the respondent would borrow money
2: Those to whom the respondent gives advice
3: Those from whom the respondent gets advice
4: Those from whom the respondent would borrow material goods
5: Those to whom the respondent would lend material goods
6: Those to whom the respondent would lend money
7: Those from whom the respondent receives medical advice
8: Non-relatives with whom the respondent socializes
```

```
9: Kin in the village
10: Those whom the respondent goes to pray with
11: Those who visit the respondent's home
12: Those whose homes the respondent visits
```

Gossip Villages (Banerjee et al., 2019) The 70 multilayer networks from villages in Karnataka, India, were generated from asking individuals the following seven questions, each of which corresponds to a layer in the resulting network.

```
1. Whose house do you go to in your free time?
2. Who comes to your house in their free time?
3. If you urgently needed kerosene, rice other groceries or money, who do you
    borrow them from?
4. Who comes to your house if he or she needed to borrow kerosene, rice, other
     groceries or money?
5. Who do you ask for advice on matters pertaining to health/finance/farming?
6. Who asks you for advice on matters pertaining to health/finance/farming?
7. Besides people living in your household, state names of your relatives who
    are living in this village.
```

References

Izabel Aguiar and Johan Ugander. The latent cognitive structures of social networks. *Network Science*, pages 1–32, 2024.

Izabel P Aguiar. Transcription of the 21 adjacency matrices in the appendix of Krackhardt's 1987 "cognitive social structures", Jun 2021. URL `https://github.com/izabelaguiar/krackhardt/`.

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning research*, 9(Sep):1981–2014, 2008.

Esraa Al-Sharoa, Mahmood Al-Khassaweneh, and Selin Aviyente. Tensor based temporal and multilayer community detection for studying brain dynamics during resting state fMRI. *IEEE Transactions on Biomedical Engineering*, 66(3):695–709, 2018.

Kristen M Altenburger and Johan Ugander. Which node attribute prediction task are we solving? within-network, across-network, or across-layer tasks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 38–48, 2021.

Arash A Amini and Elizaveta Levina. On semidefinite relaxations for the block model. *The Annals of Statistics*, 46(1):149–179, 2018.

Brett W Bader, Richard A Harshman, and Tamara G Kolda. Temporal analysis of semantic graphs using ASALSAN. In *Seventh IEEE International Conference on Data Mining*, pages 33–42. IEEE, 2007.

Brian Ball, Brian Karrer, and Mark EJ Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 2011.

Abhijit Banerjee, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. The diffusion of microfinance. *Science*, 341(6144):1236498, 2013.

Abhijit Banerjee, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. Using gossips to spread information: Theory and evidence from two randomized controlled trials. *The Review of Economic Studies*, 86(6):2453–2490, 2019.

Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. *Physical Review E*, 89(3):032804, 2014.

Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

R. Breiger, S. Boorman, and P. Arabic. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of Mathematical Psycholoy*, 12:328–383, 1975.

Jane Carlen, Jaume de Dios Pont, Cassidy Mentus, Shyr-Shea Chang, Stephanie Wang, and Mason A Porter. Role detection in bicycle-sharing networks using multilayer stochastic block models. *Network Science*, 10(1):46–81, 2022.

J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35 (3):283–319, 1970.

Yunxiao Chen, Irini Moustaki, and Haoran Zhang. A note on likelihood ratio tests for models with latent variables. *Psychometrika*, 85(4):996–1012, 2020.

Eric C Chi and Tamara G Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, 2012.

Andrzej Cichocki and Rafal Zdunek. Multilayer nonnegative matrix factorization using projected gradient approaches. *International Journal of Neural Systems*, 17(06):431–446, 2007.

Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

Caterina De Bacco, Eleanor A Power, Daniel B Larremore, and Cristopher Moore. Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E*, 2017.

Manlio De Domenico. Datasets released for reproducibility, 2022. URL `https://manliodedomenico.com/data.php`.

Manlio De Domenico and Jacob Biamonte. Spectral entropies as information-theoretic tools for complex network comparison. *Physical Review X*, 6(4):041062, 2016.

Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.

Manlio De Domenico, Albert Solé-Ribalta, Sergio Gómez, and Alex Arenas. Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences*, 111(23):8351–8356, 2014.

Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature Communications*, 6(1):1–9, 2015.

Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. Heterogeneous network representation learning. In *IJCAI*, volume 20, pages 4861–4867, 2020.

Cédric Févotte and A Taylan Cemgil. Nonnegative matrix factorizations as probabilistic inference in composite models. In *2009 17th European Signal Processing Conference*, pages 1913–1917. IEEE, 2009.

Kelly R Finn, Matthew J Silk, Mason A Porter, and Noa Pinter-Wollman. The use of multilayer network analysis in animal behaviour. *Animal Behaviour*, 149:7–22, 2019.

David N Fisher and Noa Pinter-Wollman. Using multilayer network analysis to explore the temporal dynamics of collective behavior. *Current Zoology*, 67(1):71–80, 2021.

Riccardo Gallotti and Marc Barthelemy. The multilayer temporal network of public transport in Great Britain. *Scientific Data*, 2(1):1–8, 2015.

Laetitia Gauvin, André Panisson, and Ciro Cattuto. Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach. *PloS one*, 9(1):e86028, 2014.

Amir Ghasemian, Homa Hosseinmardi, Aram Galstyan, Edoardo M Airoldi, and Aaron Clauset. Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117(38):23393–23400, 2020.

Prem Gopalan, Jake M Hofman, and David M Blei. Scalable recommendation with poisson factorization. *arXiv preprint arXiv:1311.1704*, 2013.

Richard A Harshman. Foundations of the PARAFAC procedure: Models and conditions for an explanatory multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16: 1–84, 1970.

Richard A Harshman. Models for analysis of asymmetrical relationships among N objects or stimuli. In *First Joint Meeting of the Psychometric Society and the Society of Mathematical Psychology, Hamilton, Ontario, 1978*, 1978.

Richard A Harshman and Margaret E Lundy. Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/CANDECOMP. *Psychometrika*, 61(1):133–154, 1996.

Le Thi Khanh Hien and Nicolas Gillis. Algorithms for nonnegative matrix factorization with the Kullback–Leibler divergence. *Journal of Scientific Computing*, 87(3):1–32, 2021.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Ta-Chu Kao and Mason A Porter. Layer communities in multiplex networks. *Journal of Statistical Physics*, 173(3):1286–1302, 2018.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

Hiroyuki Kasai. Stochastic variance reduced multiplicative update for nonnegative matrix factorization. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6338–6342, 2018.

Yong-Deok Kim and Seungjin Choi. Nonnegative Tucker decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. Tensorly: Tensor learning in python. *arXiv preprint arXiv:1610.09555*, 2016.

David Krackhardt. Cognitive social structures. *Social Networks*, 9(2):109–134, 1987.

Daniel B Larremore, Aaron Clauset, and Caroline O Buckee. A network approach to analyzing highly recombinant malaria parasite genes. *PLoS Computational Biology*, 9 (10):e1003268, 2013.

Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13, 2000.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

Himanshu Nayar, Benjamin A Miller, Kelly Geyer, Rajmonda S Caceres, Steven T Smith, and Raj Rao Nadakuditi. Improved hidden clique detection by optimal linear fusion of multiple adjacency matrices. In *2015 49th Asilomar Conference on Signals, Systems and Computers*, pages 1520–1524. IEEE, 2015.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning*, volume 11, pages 809–816, 2011.

Jan Overgoor, Austin Benson, and Johan Ugander. Choosing to grow a graph: modeling network formation as discrete choice. In *The World Wide Web Conference*, pages 1409–1420, 2019.

Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2): 111–126, 1994.

Subhadeep Paul and Yuguo Chen. Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel. *Electronic Journal of Statistics*, 10 (2):3807–3870, 2016.

Eleanor A Power. Social support networks and religiosity in rural South India. *Nature Human Behaviour*, 1(3):1–6, 2017.

Miklos Racz and Anirudh Sridhar. Correlated stochastic block models: Exact graph matching with applications to recovering communities. *Advances in Neural Information Processing Systems*, 34, 2021.

Fritz Jules Roethlisberger and William J Dickson. *Management and the Worker*, volume 5. Psychology Press, 1939.

Karl Rohe and Muzhe Zeng. Vintage factor analysis with varimax performs statistical inference. *arXiv preprint arXiv:2004.05387*, 2020.

Samuel F Sampson. *Crisis in a cloister*. PhD thesis, Ph. D. Thesis. Cornell University, Ithaca, 1969.

Aaron Schein, John Paisley, David M Blei, and Hanna Wallach. Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1045–1054, 2015.

Aaron Schein, Mingyuan Zhou, David Blei, and Hanna Wallach. Bayesian poisson Tucker decomposition for learning the structure of international relations. In *International Conference on Machine Learning*, pages 2810–2819. PMLR, 2016.

Asher Spector, Emmanuel Candes, and Lihua Lei. A discussion of Tse and Davidson (2022) "A note on universal inference". *Stat*, 2023.

Natalie Stanley, Saray Shai, Dane Taylor, and Peter J Mucha. Clustering network layers with the strata multilayer stochastic block model. *IEEE transactions on network science and engineering*, 3(2):95–105, 2016.

David Strieder and Mathias Drton. On the choice of the splitting ratio for the split likelihood ratio test. *Electronic Journal of Statistics*, 16(2):6631–6650, 2022.

Jimeng Sun, Spiros Papadimitriou, Ching-Yung Lin, Nan Cao, Shixia Liu, and Weihong Qian. Multivis: Content-based social network exploration through multi-way visual analysis. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 1064–1075. SIAM, 2009.

Yuchen Sun and Kejun Huang. Volume-regularized nonnegative Tucker decomposition with identifiability guarantees. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

Marc Tarrés-Deulofeu, Antonia Godoy-Lorite, Roger Guimera, and Marta Sales-Pardo. Tensorial and bipartite block models for link prediction in layered networks and temporal networks. *Physical Review E*, 99(3):032307, 2019.

Dane Taylor, Saray Shai, Natalie Stanley, and Peter J Mucha. Enhanced detectability of community structure in multilayer networks through layer aggregation. *Physical review letters*, 116(22):228301, 2016.

Dane Taylor, Rajmonda S Caceres, and Peter J Mucha. Super-resolution community detection for layer-aggregated multilayer networks. *Physical Review X*, 7(3):031056, 2017.

Dane Taylor, Mason A Porter, and Peter J Mucha. Tunable eigenvector-based centralities for multiplex and temporal networks. *Multiscale Modeling & Simulation*, 19(1):113–147, 2021.

Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 242–264. IGI global, 2010.

Timmy Tse and Anthony C Davison. A note on universal inference. *Stat*, 2022.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Toni Valles-Catala, Francesco A Massucci, Roger Guimera, and Marta Sales-Pardo. Multilayer stochastic block models reveal the multilayer structure of complex networks. *Physical Review X*, 6(1):011036, 2016.

Dingjie Wang and Xiufen Zou. A new centrality measure of nodes in multilayer networks under the framework of tensor computation. *Applied Mathematical Modelling*, 54:46–63, 2018.

Miaoyan Wang and Yuchen Zeng. Multiway clustering via tensor block models. *Advances in neural information processing systems*, 32, 2019.

Larry Wasserman, Aaditya Ramdas, and Sivaraman Balakrishnan. Universal inference. *Proceedings of the National Academy of Sciences*, 117(29):16880–16890, 2020.

Harrison C White, Scott A Boorman, and Ronald L Breiger. Social structure from multiple networks. I. blockmodels of roles and positions. *American Journal of Sociology*, 81(4): 730–780, 1976.

Samuel S Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938.

Peter Wills and François G Meyer. Metrics for graph comparison: a practitioner's guide. *PloS one*, 15(2):e0228728, 2020.

Mincheng Wu, Shibo He, Yongtao Zhang, Jiming Chen, Youxian Sun, Yang-Yu Liu, Junshan Zhang, and H Vincent Poor. A tensor-based framework for studying eigenvector multicentrality in multilayer networks. *Proceedings of the National Academy of Sciences*, 116(31):15407–15413, 2019.

Deniz Yenigun, Gunes Ertan, and Michael Siciliano. *cssTools: Cognitive Social Structure Tools*, 2016. URL `https://CRAN.R-project.org/package=cssTools`. R package version 1.0.

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40 (4):2266–2292, 2012.

Guoxu Zhou, Andrzej Cichocki, Qibin Zhao, and Shengli Xie. Efficient nonnegative tucker decompositions: Algorithms and uniqueness. *IEEE Transactions on Image Processing*, 24(12):4990–5003, 2015.