Network Science **9** (1): 99–122, 2021
doi:10.1017/nws.2020.39

CAMBRIDGE UNIVERSITY PRESS

ORIGINAL ARTICLE

Analysis of population functional connectivity data via multilayer network embeddings

James D. Wilson[1,5*], Melanie Baybay[2], Rishi Sankar[3], Paul Stillman[4] and Abbie M. Popa[5]

[1]Department of Mathematics and Statistics, University of San Francisco, San Francisco, CA 94117, USA, [2]Department of Computer Science, University of San Francisco, San Francisco, CA 94117, USA (*e-mail:* mbaybay@dons.usfca.edu), [3]Department of Computer Science, University of California, Los Angeles, CA 90095, USA (*e-mail:* rishi.sankar@gmail.com), [4]Department of Marketing, Yale School of Management, New Haven, CT 06511, USA (*e-mail:* paul.stillman@yale.edu) and [5]The Data Institute, University of San Francisco, San Francisco, CA 94117, USA (*e-mail:* apopa@gmail.com)
*Corresponding author. Email: jdwilson4@usfca.edu

Action Editor: Filippo Menczer

Abstract

Population analyses of functional connectivity have provided a rich understanding of how brain function differs across time, individual, and cognitive task. An important but challenging task in such population analyses is the identification of reliable features that describe the function of the brain, while accounting for individual heterogeneity. Our work is motivated by two particularly important challenges in this area: first, how can one analyze functional connectivity data over populations of individuals, and second, how can one use these analyses to infer group similarities and differences. Motivated by these challenges, we model population connectivity data as a multilayer network and develop the multi-node2vec algorithm, an efficient and scalable embedding method that automatically learns continuous node feature representations from multilayer networks. We use multi-node2vec to analyze resting state fMRI scans over a group of 74 healthy individuals and 60 patients with schizophrenia. We demonstrate how multilayer network embeddings can be used to visualize, cluster, and classify functional regions of the brain for these individuals. We furthermore compare the multilayer network embeddings of the two groups. We identify significant differences between the groups in the default mode network and salience network—findings that are supported by the triple network model theory of cognitive organization. Our findings reveal that multi-node2vec is a powerful and reliable method for analyzing multilayer networks. Data and publicly available code are available at https://github.com/jdwilson4/multi-node2vec.

Keywords: multilayer networks; network embedding; node2vec; Skip-gram; functional connectivity; imaging; network neuroscience

1. Introduction

Human cognition is an emergent phenomenon of complex interactions among many different brain regions (Bressler & Menon, 2010; Medaglia et al., 2015; Sporns, 2011, 2014). Network neuroscience is a common perspective of the brain in which neural connectivity is characterized through network-based models. Such network investigations have revealed general organizing principles of the whole brain, including high modularity (Sporns & Betzel, 2016), a "rich-club" of interconnected hub regions (van den Heuvel & Sporns, 2011; van den Heuvel et al., 2012), and topologies that demonstrate small-world structure (Bassett & Bullmore, 2006; Achard et al., 2006; Bassett et al., 2006; He et al., 2007). These findings have shown, for instance, that the regions of the brain not only exhibit strong clustering but also enable the brain to minimize wiring costs while

maintaining robust transfer and integration of information across regions (Bullmore & Sporns, 2012; Fornito et al., 2011). Network investigations have also advanced our understanding of neural processes, such as learning and memory (Bassett et al., 2011, 2015), cognitive control (Cole et al., 2012), and emotion (Kinnison et al., 2012). Investigations of local subnetwork structure of the brain have revealed consistent architectures that may describe overall functional efficiency (Stillman et al., 2017, 2019). Several large-scale projects have arisen from network neuroscience, such as the Human Connectome Project (Van Essen et al., 2012, 2013), as well as the BRAIN initiative (Insel et al., 2013).

Despite the many successes of network neuroscience in understanding the structure and function of the brain, many challenges remain. Our work is motivated by two particularly important challenges: (1) how can one analyze functional connectivity data over *populations* of individuals and (2) how can one utilize these analyses to infer group similarities and differences. To answer these two questions, we propose analyzing multilayer networks that effectively model the functional organization of each group. Population data of functional connectivity give rise to brain networks that are inherently multilayer—they vary across time, across person, and across cognitive task (Betzel & Bassett, 2017). Unfortunately, many network neuroscience strategies are static and consider only a single-layered network representation of the brain. Single-layered analyses neglect heterogeneity among individuals as well as their interdependencies (see Wilson et al., 2017a for a discussion). Multilayer network representations of the brain enable researchers to fully analyze the relationships within and between networks observed over time, person, or task (Bassett et al., 2011, 2015).

Multilayer networks model the functional connections between regions of the brain across a population of individuals. Multilayer networks themselves are challenging data objects to analyze, and there is a lot of current research devoted to handling these challenges (see Kivelä et al., 2014 for a recent survey). In this paper, we propose a fast and scalable algorithm, called *multi-node2vec*, that learns the nodal features from complex multilayer networks through the Skip-gram neural network model. By embedding multilayer networks of the brain to nodal features, we enable the direct analysis of the regions of the brain that are representative of the group under study.

We apply multi-node2vec to a multilayer brain network representing the functional connectivity of 74 healthy individuals and 70 patients with schizophrenia who underwent resting state fMRI. We demonstrate how to utilize the results of multi-node2vec for three primary objectives: (i) visualization and clustering of these regions into communities of similar features, (ii) classification of regions into anatomical regions of interest (ROIs) in the brain, and (iii) comparing two populations of individuals. We find that multi-node2vec identifies feature embeddings that closely match the functional organization of healthy individuals and also provides a powerful strategy for comparing groups of individuals. Our proposed embedding technique provides a valuable step in automatically learning neurological variation among brains, including individual differences and disease.

1.1 Related work

Feature engineering is a common and important learning task in statistics and machine learning. Traditionally, feature engineering for networks, often referred to as network embedding, has amounted to manually describing summaries of networks based on a collection of user-selected network properties, like structural importance or subgraph counts (Gallagher & Eliassi-Rad, 2010; Henderson et al., 2011). A similar strategy has been applied to multilayer networks, where chosen features attempt to quantify the within- and between-layer relationships among nodes (Boccaletti et al., 2014; Kivelä et al., 2014). In contrast to these approaches, the multi-node2vec algorithm automatically learns important continuous features of multilayer networks and requires no user input on what properties to capture.

Feature engineering techniques have been extensively used to identify low-rank representations of multivariate data. In this setting, the data matrix $\mathbf{X}$ is an $n \times p$ matrix whose rows are

n independent observations measured on p features. Dimension reduction techniques are particularly important when the data are high dimensional—when $p > n$—as traditional statistical inference is often no longer reliable (Bühlmann & Van De Geer, 2011). Singular value decompositions, principal components analysis, and spectral clustering, for instance, are well-studied decomposition techniques that have been applied to a number of high-dimensional problems ranging from topic modeling to micro-array analysis. These techniques rely on the spectral decomposition of $\mathbf{X}$, its empirical covariance matrix, and the graph Laplacian of a similarity matrix on the columns of $\mathbf{X}$, respectively. Though these methods are known to provide accurate representations of $\mathbf{X}$, they face a drawback in computational complexity for large p due to matrix inversion, which can be prohibitive for especially high-dimensional problems. In Section 6, we show that multi-node2vec is in fact an approximation to closed-form implicit matrix factorization.

There have been many feature learning techniques for static networks developed in the past decade. The latent space model from Hoff et al. (2002), for example, is a common model-based embedding technique that embeds the observed network onto Euclidean space—typically onto two dimensions. Our current work is most closely related to the automatic feature learning techniques LINE (Tang et al., 2015), DeepWalk (Perozzi et al., 2014), and node2vec (Grover & Leskovec, 2016). We briefly discuss these here but refer the reader to Goyal & Ferrara (2018) for a recent review of node embedding techniques for static networks. LINE, DeepWalk, and node2vec each learn features of a node from the neighborhoods of the node in the observed graph. LINE learns d-dimensional features by an objective function that preserves first- and second-order network properties. DeepWalk and node2vec each learn D-dimensional features using the Skip-gram neural network model, which minimizes a log-likelihood loss function that characterizes relationships from node neighborhoods in the observed graph. The Skip-gram model was originally developed for learning efficient representations of words in a large document of text (Mikolov et al., 2013b; Pennington et al., 2014). The first application of the Skip-gram model was in the word2vec algorithm (Mikolov et al., 2013a), where it was used to estimate a word's features through the log-likelihood cost minimization from the prediction of that word's context. DeepWalk, node2vec, and multi-node2vec differ in the way they collect node neighborhoods. DeepWalk extracts neighborhoods using truncated random walks. Node2vec performs second random walks based on hyperparameters that guide the likelihood of visiting nodes either closer to or further away from previously visited nodes. The development of our algorithm is motivated by the recent success of the node2vec algorithm on consensus matrices of structural magnetic resonance imaging, (Rosenthal et al., 2018). The multi-node2vec algorithm, however, directly handles the analysis of populations of functional connectivity data. Multi-node2vec is also random walk based and can be thought of as a generalization of the original DeepWalk and node2vec algorithms. Utilizing Laplacian dynamics like that discussed in Mucha et al. (2010), we incorporate a walk parameter that dictates the probability of moving from one layer to the next.

Other recent work has focused on generative network models that model populations of networks, including the random effects stochastic block model (Paul & Chen, 2018), the multi-subject stochastic block model (Pavlovic et al., 2019), the hierarchical latent space model (Wilson et al., 2020), as well as the edge-based logistic model from Simpson et al. (2019). These three models each assume independence of the edges within and across individuals. Even so, estimation methods for these models are sometimes prohibitive and typically require small network representations (on the order of 10 s of nodes) for each individual.

The community detection task of partitioning the nodes of a multilayer network into densely connected subgroups, or communities, can be viewed as multilayer embedding. Specifically, the results of a community detection algorithm is an $N \times D$ matrix $\mathbf{F}$, where the vth row $\mathbf{f}_v$ is a binary vector that indicates which community(ies) the node v is contained. The development of multilayer community detection methods is still in its early stages, but several useful techniques have been developed over the past decade (De Domenico et al., 2015; Mucha et al., 2010; Stanley et al., 2016; Wilson et al., 2017a). Though not the focus of this paper, it would be interesting to fully

explore the use of communities as features for regression and other machine learning tasks in future work.

2. Multilayer embedding with multi-node2vec

In resting state functional connectivity, network models are constructed by gauging the degree to which two regions' time series activity is related to one another. The intuition is that the greater two regions are functionally connected, the more their time series should co-activate. In the present study, we model the strength of connection between two regions based on the correlation between the two regions' activity during a resting state fMRI scan (i.e., when participants have no task except to stay awake (Bullmore & Sporns, 2009; Smith et al., 2011). We can subsequently apply the multi-node2vec technique to identify local features of the brain from a group of individuals.

A multilayer network of length m is a collection of networks or graphs $\{G_1, \ldots, G_m\}$, where the graph G_ℓ models the relational structure of the ℓth layer of the network. Each layer $G_\ell = (V_\ell, W_\ell)$ is described by the vertex set V_ℓ that describes the units, or actors, of the layer, and the *intra-layer* edge weights $W_\ell = \{w_\ell(u, v) : u, v \in V_\ell\}$ that describes the strength of relationship between the nodes. Furthermore, there is a collection of *inter-layer* edge weights $IL := \{w_{\ell,\ell'}(u, v) : u \in V_\ell, v \in V_{\ell'}\}$ that describe relationships between nodes of differing layers. Note that layers in the multilayer sequence may be heterogeneous across vertices, edges, and size. In the case of population studies of functional connectivity, each layer G_ℓ represents the correlation network arising from resting state fMRI for individual ℓ. Denote the set of unique nodes in $\{G_1, \ldots, G_m\}$ by $\mathcal{N}$, and let $N = |\mathcal{N}|$ denote the number of nodes in that set. Throughout the remainder of this paper, to signify the unique node set $\mathcal{N}$, we represent multilayer networks with m layers and node set $\mathcal{N}$ as $\mathbf{G}^m_{\mathcal{N}}$.

Multilayer networks are inherently complex and high dimensional. Without further assumptions on $\mathbf{G}^m_{\mathcal{N}}$, inference on $\mathcal{N}$ necessitates the modeling of N^2 (possibly dependent) edge variables, which is computationally challenging even for moderately sized N. In light of this challenge, the aim of the current work is to learn an interpretable low-dimensional feature representation of the nodes in a multilayer network. In particular, we seek a D-dimensional representation:

$$\mathbf{F} : \mathcal{N} \to \mathbb{R}^D \tag{1}$$

where $D << N$. The function $\mathbf{F}$ can be viewed as an $N \times D$ matrix whose rows $\{\mathbf{f}_v : v = 1, \ldots, N\}$ represent the feature space of each node in $\mathcal{N}$.

2.1 Maximum likelihood formulation

Let $\mathbf{G}^m_{\mathcal{N}}$ be an observed multilayer network with m layers and the set of unique nodes $\mathcal{N}$. Our aim is to learn D representative features of $\mathcal{N}$ given by the matrix $\mathbf{F}$ in Equation (1). This learning task can be formulated as a problem of maximum likelihood estimation. To see this, one can view $\mathbf{G}^m_{\mathcal{N}}$ as a realization of a random graph on the node set $\mathcal{N}$ whose joint probability distribution is dictated by the feature matrix $\mathbf{F}$. We calculate an estimator for $\mathbf{F}$ in Equation (1), $\widehat{\mathbf{F}}$, that maximizes the joint likelihood:

$$\mathbb{L}(\mathbf{F} \mid \mathbf{G}^m_{\mathcal{N}}) = \mathbb{P}(\mathbf{G}^m_{\mathcal{N}} \mid \mathbf{F}) \tag{2}$$

where $\mathbb{P}$ is the joint distribution of a multilayer graph with m layers and unique node set $\mathcal{N}$ given the feature representation $\mathbf{F}$. In general, maximization of Equation (2) is computationally intractable. We therefore make two simplifying assumptions about the joint distribution $\mathbb{P}$. Our

assumptions rely upon a suitable definition of a *multilayer neighborhood*. Defining the neighborhood of a node is related to the problem of defining the context of a word in a large document from natural language processing. In static unweighted networks, the neighborhood of the node u is often defined as the collection of nodes that share an edge with u. This definition is motivated by the homophily principle (McPherson et al., 2001), which posits that nodes with similar features are highly connected to one another in the network. In many cases, this definition of a neighborhood is restrictive. This is particularly true when the observed network is only partially observed or when the edges of the network are generated from some underlying noisy process. We instead define a multilayer neighborhood of node u based on a dynamic process across the network. Our construction is a generalization of the random walk constructions from Grover & Leskovec (2016), Perozzi et al. (2014), and Tang et al. (2015) and is analogous to the defining of communities via Laplacian dynamics as in Lambiotte et al. (2008) and Mucha et al. (2010). To be specific, we define the neighborhood of node u as the collection of vertices that are visited over a random walk on the multilayer network $\mathbf{G}_{\mathcal{N}}^{m}$. We make this more formal below when describing the neighborhood search procedure of the algorithm.

Once the multilayer neighborhood of each node has been defined, we make two simplifying assumptions given the feature matrix $\mathbf{F}$. First, we assume that the joint distribution characterizing $\mathbf{G}_{\mathcal{N}}^{m}$ is the same as the distribution characterizing the collection of neighborhoods in $\mathbf{G}_{\mathcal{N}}^{m}$. This assumption is reasonable if we believe that the features $\mathbf{F}$ provide the same information as the multilayer network itself. Second, given the feature matrix $\mathbf{F}$, we assume that the neighborhood of a node v depends only on its own feature representation, $\mathbf{f}_v$ and given this representation is independent of the neighborhoods of other nodes $u \in \mathcal{N}$. These assumptions are the same as those made for the node2vec algorithm for static networks (Grover & Leskovec, 2016) and are analagous to those made for word2vec, which assumes that the joint probability distribution of a collection of text can be characterized by the distribution of the collection of conditionally independent word contexts given each word's feature representation (Mikolov et al., 2013b).

With the conditional independence assumptions of the neighborhoods given $\mathbf{F}$, maximizing the joint likelihood of $\mathbf{F}$ given the entire network $\mathbf{G}_{\mathcal{N}}^{m}$ reduces to the task of identifying the features $\mathbf{f}_v$ given the neighborhood of v in $\mathbf{G}_{\mathcal{N}}^{m}$. Let $Ne(u)$ denote the neighborhood of node u, namely the collection of nodes that are linked to u. Given the neighborhood of each node, the likelihood from Equation (2) simplifies to:

$$\mathbb{L}(\mathbf{F} \mid \mathbf{G}_{\mathcal{N}}^{m}) = \prod_{u \in \mathcal{N}} \mathbb{P}\left(\mathrm{Ne}(u) \mid \mathbf{f}_u\right) \tag{3}$$

As it remains a challenging task to quantify the dependence between the neighborhoods of differing layers, the maximization of Equation (3) is still computationally difficult. Thus, we define a family of multilayer graphs for which this maximization is feasible. It turns out that we can define such a family by assuming minimal conditional independence assumptions given the representation $\mathbf{F}$, described as follows. Let $\mathcal{G}_{\mathcal{N}}^{m}$ denote the family of multilayer graphs whose members are random graphs with m layers and unique nodes $\mathcal{N}$. For every member of $\mathcal{G}_{\mathcal{N}}^{m}$, assume that the following hold

(A1) For all $u \in \mathcal{N}$, $\mathbb{P}(\mathrm{Ne}(u) \mid \mathbf{f}_u) = \prod_{v \in \mathrm{Ne}(u)} \mathbb{P}(v \mid \mathbf{f}_u)$

(A2) Let $u \in \mathcal{N}$. For every $v \in \mathrm{Ne}(u)$, $\mathbb{P}(v \mid \mathbf{f}_u) = \mathbb{P}(u \mid \mathbf{f}_v)$.

Assumption (A1) characterizes the local conditional independence among nodes in the neighborhoods of a node v given its feature representation, $\mathbf{f}_v$. Assumption (A2) enforces a symmetric effect of neighboring nodes in their feature space. A consequence of (A2) is that for any node v that is a neighbor of u, the following relationship holds

$$\mathbb{P}(v \mid \mathbf{f}_u) = \frac{\exp\{\mathbf{f}_v^T \mathbf{f}_u\}}{\sum_{w \in \mathcal{N}} \exp\{\mathbf{f}_w^T \mathbf{f}_u\}}$$

If the observed graph $\mathbf{G}_{\mathcal{N}}^m$ is a realization of a multilayer random graph from the family $\mathcal{G}_{\mathcal{N}}^m$ under which assumptions (A1) and (A2) hold, then Equation (3) can be expressed as:

$$\mathbb{L}(\mathbf{F} \mid \mathbf{G}_{\mathcal{N}}^m) = \prod_{u \in \mathcal{N}} \prod_{v \in \mathrm{Ne}(u)} \frac{\exp\{\mathbf{f}_v^T \mathbf{f}_u\}}{\sum_{w \in \mathcal{N}} \exp\{\mathbf{f}_w^T \mathbf{f}_u\}} \tag{4}$$

Maximizing Equation (4) is equivalent to maximizing the following log-likelihood:

$$\mathcal{L}(\mathbf{F} \mid \mathbf{G}_{\mathcal{N}}^m) = \sum_{u \in \mathcal{N}} \sum_{v \in \mathrm{Ne}(u)} \left[\mathbf{f}_v^T \mathbf{f}_u - \log\left(Z_u\right) \right] \tag{5}$$

where $Z_u = \sum_{w \in \mathcal{N}} \exp\{\mathbf{f}_w^T \mathbf{f}_u\}$ is a normalization constant for the node u. Following the approach of Grover & Leskovec (2016) and Mikolov et al. (2013b), we approximate Z_u using negative sampling. We note, however, that Markov chain Monte Carlo sampling methods could also be used to approximate Z_u as in Wilson et al. (2017b) and Denny et al. (2017). The use of Skip-gram with negative sampling is appealing for two reasons: (i) the algorithm is fast and scalable to large multilayer networks and (ii) the strategy is closely related to matrix factorization (Levy & Goldberg, 2014; Qiu et al., 2018) as we will see in Section 5.

Given an observed multilayer network $\mathbf{G}_{\mathcal{N}}^m$, multi-node2vec is an approximate algorithm that estimates $\mathbf{F}$ through maximization of the log-likelihood function in Equation (5). The algorithm consists of two key steps. First, the NeighborhoodSearch procedure identifies a collection of s neighborhoods of length l for $\mathbf{G}_{\mathcal{N}}^m$ through second-order random walks on the network. The NeighborhoodSearch procedure depends on three hyperparameters—p, q, and r—that dictate the exploration of the random walk away from the source node and the tendency to traverse layers. Once a collection of neighborhoods or *BagOfNodes* have been identified, the log-likelihood in Equation (5) is optimized in the Optimization step using stochastic gradient descent on the two-layer Skip-gram neural network model of context size k. The result of the Optimization procedure is a D-dimensional feature representation $\mathbf{F}$. Figure 1 provides an illustration. We describe the NeighborhoodSearch and Optimization procedures in more detail next.

2.2 The NeighborhoodSearch procedure

Multi-node2vec begins by parsing a multilayer network into a collection of neighborhoods for each unique node in $\mathcal{N}$. The NeighborhoodSearch procedure identifies this collection of neighborhoods, or *BagofNodes*, using s truncated second-order random walks of length l. Without loss of generality, suppose that node labels among layers are registered in the sense that node u in vertex set V_ℓ represents the same actor as node u in vertex set $V_{\ell'}$. To construct the random walk, we consider the collection of weights $\{w_{\ell,\ell'}(u,v) : \ell, \ell' \in 1, \ldots, m; u, v \in \mathcal{N}\}$, where $w_{\ell,\ell'}(u,v)$ defines the edge weight between node u from layer ℓ and node v from layer ℓ'. Thus, the collection of edge weights $\{w_{\ell,\ell'}(\cdot,\cdot) : \ell \neq \ell'\}$ represent the *inter-layer edges*, whereas, the collection $\{w_{\ell,\ell'}(\cdot,\cdot) : \ell = \ell'\}$ represent the *intra-layer edges* in the multilayer network.

For an observed multilayer network and its edge weights defined as above, the NeighborhoodSearch procedure identifies s neighborhoods using second-order random walks over the nodes and layers of length ℓ, constructed as follows. Let u_i be the ith node visited by the random walk and ℓ_i the corresponding layer. Suppose, without loss of generality, that the initial pair (u_1, ℓ_1) is chosen uniformly at random. Subsequent vertex, layer pairs are visited according to the conditional probability:

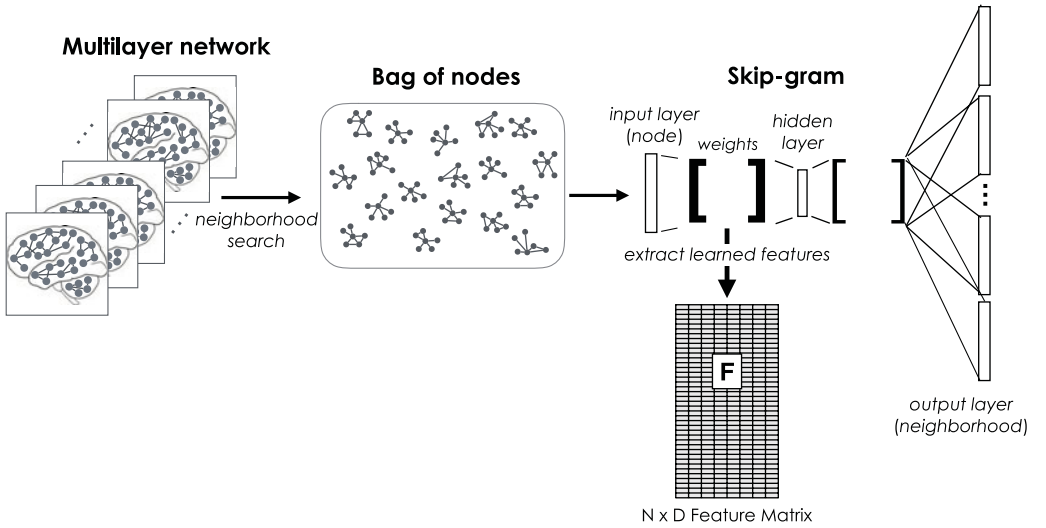

Figure 1. Demonstration of the multi-node2vec algorithm. Beginning with a multilayer network (left), one first identifies a collection of multilayer neighborhoods (bag of nodes) via the NeighborhoodSearch procedure. Next, the Optimization procedure calculates the maximum likelihood estimator **F** through the use of the Skip-gram neural network model (right) on the identified bag of nodes.

$$\mathbb{P}(u_i = x, \ell_i = \ell' \mid u_{i-1} = v, \ell_{i-1} = \ell) = \frac{\pi_{v,x,\ell,\ell'}}{Z}, \quad w_{\ell,\ell'}(v,x) > 0 \tag{6}$$

where $\pi_{v,x,\ell,\ell'}$ is the unnormalized transition probability of moving from vertex–layer pair (v, ℓ) to pair (x, ℓ'), and Z is a normalizing constant. We set $\pi_{v,x,\ell,\ell'}$ as a function of the walk parameters p, q, and r as follows:

$$\pi_{v,x,\ell,\ell'} = \alpha_{pqr}(t,x,\ell,\ell') \cdot w_{\ell,\ell'}(v,x) \tag{7}$$

The $\alpha_{pqr}(t,x,\ell,\ell')$ term acts as a search bias on the observed weights that depends on the previously traversed edge (t, v). That is, the walk now resides at node v having just traveled from node t and the next node that the random walk visits depends on (a) the distance t is from the future node and (b) whether there is a layer transition. Let $d_\ell(t,x)$ denote the shortest path distance between nodes t and x in layer ℓ. To account for layer transitions, we further decompose $\alpha_{pqr}(t,v,x,\ell,\ell')$ as:

$$\alpha_{pqr}(t,x,\ell,\ell') = \beta_{pq}(t,x)\,\mathbb{I}(\ell' = \ell) + \gamma_r(v,x)\,\mathbb{I}(\ell' \neq \ell) \tag{8}$$

where $\beta_{pq}(t,x) = p^{-1}\mathbb{I}(d_\ell(t,x) = 0) + \mathbb{I}(d_\ell(t,x) = 1) + q^{-1}\mathbb{I}(d_\ell(t,x) = 2)$ and $\gamma_r(v,x) = r^{-1}\mathbb{I}(x = v)$. The $\beta_{pq}(t,x)$ term controls the rate at which the random walk explores and leaves the neighborhood of a node within layer ℓ. This quantity is the same as that specified for static networks in node2vec and has been shown to identify neighborhoods that interpolate between outcomes of breadth first search and depth first search. The return parameter p controls the likelihood of revisiting the same node, layer pair, whereas, the in-out parameter q controls exploration of the walk in layer ℓ. The $\gamma_r(v,x)$ term controls the rate at which a random walk transitions from one layer to another. Setting the layer walk parameter r to be large ($> \max(p, q, 1)$) ensures little layer-to-layer exploration. Setting r in this way encourages independent neighborhood sampling across layers. On the other hand, setting r to be small ($< \min(p, q, 1)$) promotes exploration among layers, and the resulting neighborhoods will reflect dependency among the layers.

Once the parameters s, l, p, q, and r have been chosen, s random walks of length l are performed on the nodes of the observed multilayer network using transition probabilities from Equation (6). These s samples serve as the *BagofNodes* from which the nodal features are learned.

2.3 Optimization

For a given dimension size D, a context size k, and the collection of neighborhoods from the NeighborhoodSearch step, multi-node2vec then minimizes the cost of Equation (5) using stochastic gradient descent and the Skip-gram two-layer neural network model. For each node, the normalization constant Z_u is approximated using negative sampling. The Skip-gram model iteratively updates the matrix $\mathbf{F}$ in the following manner. Each node is encoded as a one-hot vector and provided as the input layer to a two-layer neural network from which the neighborhood of the node is predicted. Applying the log-likelihood $\mathcal{L}$ as a cost function, the error of the prediction is calculated. Partial derivatives of the cost function with respect to the rows of each of the intermediate weight matrices are calculated and updated using stochastic gradient descent to minimize cost. This procedure is repeated across all nodes in $\mathcal{N}$ until the cost function can no longer be reduced. After learning from each of the neighborhoods in our bag of nodes, we extract the model's *node embeddings*—the $N \times D$ representation weight matrix associated with Skip-gram's input layer.

This optimization is analogous to that of the node2vec algorithm, but in our application the weight matrices of the two-layer neural network are D-dimensional representations of the unique nodes $\mathcal{N}$ and thus account for the dependence among layers in the multilayer network. It should be noted that multi-node2vec is an approximate algorithm that relies upon the normalizing constants $\{Z_u\}$, as well as the approximate optimization of stochastic gradient descent. Though not the focus of this paper, there has been a lot of recent work investigating the optimality landscape of gradient descent methods (see e.g., Lee et al., 2016), which provides promising theoretical justification for its use.

The choice of k directly affects the amount of information one gains for each node but its value depends on the sparsity of the observed network. Large values of k introduce undesired noise to the identified neighborhoods, whereas, values of k that are too small result in neighborhoods that do not contain significant information about the neighborhoods in the network. We found that setting k near the average degree of the network provided the best results in our numerical studies. In the case that the observed network is either densely connected or contains few layers, the neighborhoods for each node may not contain sufficient information to inform the desired features. In such scenarios, it may be desirable to sample multiple neighborhoods for each node. Thus, we include an optional parameter a that specifies the minimum number of samples generated for each node. Unless otherwise specified, we set $a = 1$ in our numerical studies. Finally, the dimensionality parameter D should be chosen to provide sufficient information about the multilayer network while greatly reducing the total number of nodes N, though it is an open problem to understand an optimal dimension to represent general static networks.

3. Efficient implementation of multi-node2vec

Consider a multilayer functional connectivity network $\mathcal{G}_{\mathcal{N}}^{m}$ with N unique nodes and m layers and non-negative edge weights. By construction, multi-node2vec requires the storage of $O(mN^2 + Nm^2)$ different edge weights, since there are $O(mN^2)$ intra-layer edges and $O(Nm^2)$ inter-layer edges. This can quickly overwhelm computational resources when the number of layers or unique nodes is large. It turns out that multi-node2vec can be applied by only storing $O(N^2)$ values, which greatly improves the efficiency of the algorithm. We begin by analyzing the relationship of multi-node2vec with the node2vec and DeepWalk algorithms. To do so, we need a notion of equivalence between two stochastic algorithms. For this purpose, we consider the stochastic equivalence of two algorithms, defined as follows.

Definition 1. *Let A_1 and A_2 be two stochastic algorithms, each with the same set of possible outcomes Ω. That is, for fixed input data X, A_k is a random function that maps X to an outcome $o \in \Omega$:*

$A_k(X) \to o \in \Omega$. Define $\mathbb{P}_k$ as the probability mass function characterizing the probability of each possible outcome of A_k: $\{\mathbb{P}_k(A_k(X) = o) : o \in \Omega\}$. A_1 and A_2 are said to be stochastically equivalent if $\mathbb{P}_1 = \mathbb{P}_2$.

Let $\mathbb{A}$ denote the $N \times N$ aggregate adjacency matrix of the nodes $\mathcal{N}$ with entries $\mathbb{A}_{u,v} = \sum_\ell \sum_{\ell'} w_{\ell,\ell'}(u, v)$. Define the adjusted version of $\mathbb{A}$, $\widetilde{\mathbb{A}}(r)$, as the $N \times N$ matrix with entries:

$$\widetilde{\mathbb{A}}_{u,v}(r) = r^{-1} \sum_{\ell \neq \ell'} w_{\ell,\ell'}(u, v) + \sum_{\ell} w_{\ell,\ell}(u, v), \qquad u, v \in \mathcal{N}$$

Note that $\widetilde{\mathbb{A}}(r) = \mathbb{A}$ when $r = 1$. One can view the matrix $\widetilde{\mathbb{A}}(r)$ as an adjusted adjacency matrix whose edge weights depend on the layer walk parameter r. Write $\widetilde{\mathbf{G}}_{\mathcal{N}}(r)$ as the graph with nodes $\mathcal{N}$ and edge weights specified by the adjacency matrix $\widetilde{\mathbb{A}}(r)$.

The following lemma relates multi-node2vec with node2vec and DeepWalk and shows under what conditions they are stochastically equivalent in terms of the walk parameters p, q, and r.

Lemma 3.1. *Let $\mathbf{G}^m_{\mathcal{N}}$ be an observed multilayer network and let $\widetilde{\mathbf{G}}_{\mathcal{N}}(r)$ be its adjusted aggregate network. Suppose that the parameters D, k, s, l are held constant. Then, the following hold*

(a) *for all $p, q, r > 0$, the application of multi-node2vec to $\mathbf{G}^m_{\mathcal{N}}$ is stochastically equivalent to the application of node2vec to $\widetilde{\mathbf{G}}_{\mathcal{N}}(r)$;*
(b) *if $p = q = 1$, the application of multi-node2vec to $\mathbf{G}^m_{\mathcal{N}}$ is stochastically equivalent to the application of DeepWalk to $\widetilde{\mathbf{G}}_{\mathcal{N}}(r)$.*

Proof. Since multi-node2vec, node2vec, and DeepWalk all use Skip-gram on identified neighborhoods, it will suffice to show that the transition probabilities of the random walks used to identify the neighborhoods for each method are equal under the stated conditions to prove Theorem 3.1. We begin by proving part (a) for general $p, q, r > 0$. Let $\pi_{u,v}$ denote the unnormalized transition probability of the random walk traveling from $u \to v$ based on the application of node2vec on the graph $\widetilde{\mathbf{G}}_N(r)$. Similarly, let $\pi^*_{u,v}$ denote this unnormalized transition probability of the random walk based on the application of multi-node2vec to $\mathbf{G}^m_{\mathcal{N}}$. Then by the law of total probability, we have

$$\begin{aligned}\pi^*_{u,v} &:= Z \cdot P_{\mathcal{G}^m_{\mathcal{N}}}(u_{j+1} = u \mid u_j = v) = Z \cdot \sum_{\ell} \sum_{\ell'} w_{\ell,\ell'}(v, x) P(\ell_{i-1} = \ell) \\ &= \beta_{pq}(t, v) \sum_{\ell} w_{\ell,\ell}(u, v) + \gamma_r(u, x) \sum_{\ell \neq \ell'} w_{\ell,\ell'}(u, v).\end{aligned}$$

Note that $\beta_{pq}(t, v) = 1$ when $v = u$ and that $\gamma_r(u, x) = r^{-1}$. It follows that $\pi^*_{u,v} = \pi_{u,v}$ and thus part (a) is proved. Part (b) is proven in an analogous fashion by taking $\pi_{u,v}$ as the transition probability for the random walk associated with DeepWalk on the graph $\widetilde{\mathbf{G}}_N(r)$ and noting that $\beta_{pq}(t, v) \equiv 1$ when $p = q = 1$. □

Lemma 3.1 reveals that the application of multi-node2vec on an observed multilayer network $\mathbf{G}^m_{\mathcal{N}}$ is stochastically equivalent to the application of node2vec on the adjusted aggregate graph $\widetilde{\mathbf{G}}_{\mathcal{N}}(r)$. In practice, this means that running multi-node2vec on an observed multilayer network will provide the same results as running node2vec on the corresponding adjusted aggregate network if the same seed set is specified for a random number generator. This suggests that multi-node2vec can be implemented with just the storage of $\widetilde{A}(r)$, which contains $O(N^2)$ edge weights. In the special case that $p = q = 1$, one can equivalently run multi-node2vec, node2vec, or DeepWalk.

Table 1. The number of regions in each functional subnetwork of the whole brain when the Power atlas parcellation is applied to each whole-brain network. These subnetworks are used to demonstrate the use of multi-node2vec in the classification study in Section 4.3

Auditory: 13	Dorsal attention: 11	Default mode: 58
Salience: 15	Memory/retrieval: 5	Frontoparietal task control: 25
Visual: 31	Ventral attention: 9	Subcortical: 13
Cerebellar: 4	Uncertain: 28	Cingulo-opercular task control: 14
Sensory—Hand: 30	Sensory—Mouth: 5	

4. Numerical study

We now apply multi-node2vec to a multilayer brain network representing the functional connectivity of 74 healthy individuals and 60 patients with schizophrenia who underwent resting state fMRI. In this case study, we demonstrate the use of multi-node2vec for three primary objectives: (i) clustering of brain regions into communities of similar features, (ii) classification of nodes into anatomical ROIs in the brain, and (iii) comparing and classifying two populations of individuals. To assess overall performance, we compared multi-node2vec with several off-the-shelf embedding techniques, including LINE, DeepWalk, and node2vec. Our analysis reveals that multi-node2vec identifies features that closely associate with the functional organization of the brain and provides a powerful strategy for comparing across groups of individuals.

To analyze the efficacy of multi-node2vec, we consider the tasks of clustering and classification of of ROIs using the subnetwork labels as ground truth. We furthermore analyze multi-node2vec via a classification study, where we aim to classify healthy individuals from schizophrenia patients using global summaries of the identified multilayer embeddings. Publicly available code for the multi-node2vec algorithm as well as all code used for our findings are available at https://github.com/jdwilson4/multi-node2vec.

4.1 Description of data

We investigate a dataset of resting state fMRI scans of 74 healthy individuals (aged 18–65 years, 23 female) and 60 individuals with schizophrenia (aged 18–65 years, 16 female) from the Center for Biomedical Research Excellence (COBRE Mayer et al., 2013) posted to the 1,000 Functional Connectomes Project (Biswal et al., 2010). Participants had no history of neurological disorder, mental retardation, substance abuse, or dependencies in the last 12 months, or severe head trauma. Participants underwent 5 minutes of resting state fMRI in which they had no task except to stay awake, followed by a multi-echo MPRAGE scan (see Mayer et al., 2013 for scanning parameters and preprocessing information).

To construct the multilayer representation of this dataset, we use a previously validated atlas (Power et al., 2011) that specifies 264 spheres of radius 8 mm, which constitute our 264 ROIs. We averaged the fMRI time series from all voxels within each ROI, yielding 264 time series per participant. For each of these time series, we regressed out six motion parameters (to account for head movement), four parameters corresponding to cerebrospinal fluid, and four parameters corresponding to white matter. These steps have been shown to reduce bias and noise within the data (Chai et al., 2012). Finally, for each participant, we correlated the 264 time series with one another, yielding a 264×264 correlation matrix for each participant. We analyze the weighted multilayer network representation of these data. Intra-layer edges are encoded with a weight of $w_{\ell,\ell}(u,v)\mathbb{I}(r(u,v) > 0)$, where $r(u,v)$ is the correlation between the two incident regions. Inter-layer edges are encoded as $w_{\ell,\ell'}(u,v) = 1$ when $u = v$ and 0 otherwise. The ground-truth subnetwork labels are previously defined functional subnetworks, established in Power et al. (2011). The subnetwork labels and number of regions belonging to each functional subnetwork are presented in Table 1.

4.2 Simulation study and parameter choices

Before applying multi-node2vec to the fMRI data, we first describe a strategy for choosing the parameters for the algorithm through the use of simulation and theoretical study. In Grover & Leskovec (2016), the authors recommended values for the walk parameters p and q based on extensive empirical studies for node2vec. For consistency and ease of comparison, we use their recommended values ($p = 1, q = 0.5$) for both node2vec and multi-node2vec in our application though we mention that these too could be tuned using simulation. In Section 5, we study the limiting behavior of multi-node2vec as a function of the walk length parameter, l, and find that asymptotically in l the embeddings from multi-node2vec converge to the result of non-negative matrix factorization. To balance computational speed and theoretical gaurantees, we therefore suggest using a moderately sized l in application and opt for $l = 30$. For the layer walk parameter r, we assess the performance of three different values, $r = 0.25, .5, .75$, on the fMRI dataset to investigate differences among these values.

Through simulation, we are particularly interested in three aspects of multi-node2vec: (i) analyzing the specificity of multi-node2vec, (ii) investigating the effects of the neighborhood or context size of the identified neighborhoods, k, and the dimension of the feature vectors D as they relate to the size, structure, and connectivity of the network, and (iii) analyzing the scalability of multi-node2vec for networks with a large number of nodes and/or layers.

For (i), we simulated a multilayer graph where each layer was an independent Erdős–Rényi random graph with probability of connection set to the average degree of that group. These simulated graphs represent what a multilayer network would look like at random with no topological structure other than preserving the average degree of the group of images.

For (ii) and (iii), we use unweighted multilayer networks using a multilayer generalization of the planted partition model, designed to align with the connectivity, clustering, and size of the observed multilayer networks in our fMRI study. In all simulations, each layer of the simulated multilayer network contains $n = 264$ nodes to match the fMRI networks in our application. Nodes were placed randomly into c equally sized communities. For each layer, edges are placed randomly between two nodes of the same community with probability p_{in} and edges are placed between two nodes of differing communities with probability p_{out}. With this construction, each layer of the generated network has the same community structure across layers. This graph model is a special case of the multilayer stochastic block model (MSBM) considered (Han et al., 2015; Stanley et al., 2016; Wilson et al., 2017a). For our analysis, nodes of the same community are expected to have similar features with one another and different features than nodes from other communities. This model therefore provides a well-structured multilayer network for which we can study and tune multi-node2vec. We analyze the effect of k on the performance of the algorithm on the MSBM. We note that one could also tune the dimension parameter D through an analogous simulation study. In our application, we have access to ground-truth labels for the nodes—the functional subnetwork label—and therefore directly compare the performance of multi-node2vec against competing methods by running each method across a grid of dimension D ranging from 2 to 100.

For each of the following studies, we set $p_{in} = 0.49$ to match the average degree of the functional brain networks. To assess the relevance of the features identified by multi-node2vec, we compare the clusters obtained from the k-means algorithm on the feature matrix with the true community labels of the network and calculate the adjusted rand score as a measurement of match between the two partitions. For each simulation, we replicate the study 30 times and report the average adjusted rand score. The results for each simulation is presented in Figure 2 and discussed below.

4.2.1 Specificity of multi-node2vec

To test the specificity of the results identified in our study, we first applied multi-node2vec to multilayer Erdős–Rényi random graphs of the same size and expected degree as the populations that we investigated. We expect that the embeddings of completely random multilayer graphs would give no structural insights, and thus that the clusters identified from the embeddings would not

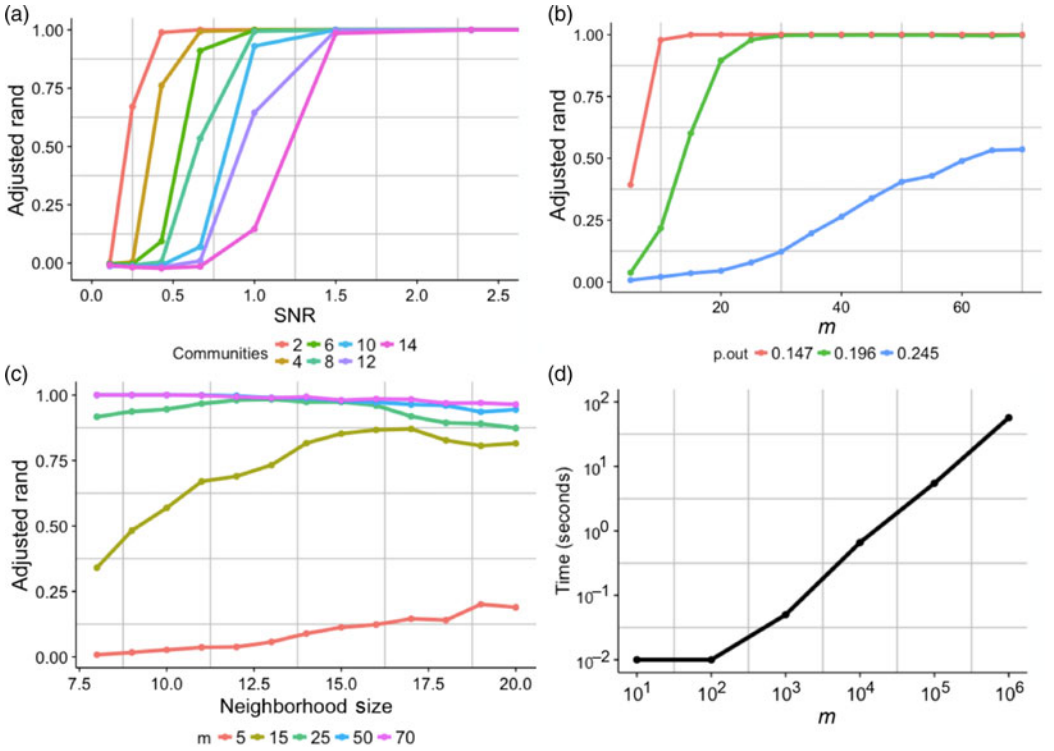

Figure 2. Simulation results from the numerical study described in Section 6. All simulations are repeated 30 times and the average is shown. (a) The adjusted rand index score of the clusters identified by k-means clustering on the identified feature matrix from multi-node2vec applied to the MSBM as a function of the signal-to-noise ratio (SNR) = $p_{in}/p_{out} - 1$ and (b) across the number of layers in the network. (c) The adjusted rand index score of the clusters identified by k-means clustering on the identified feature matrix from multi-node2vec as a function of the neighborhood size input to the algorithm. (d) The average time (in seconds) required by multi-node2vec on multilayer random graphs with 10 nodes in each layer and m layers. Notably, networks with 1 million layers required just 58 seconds.

closely align with the functional subnetworks. This test provides a validation that multi-node2vec can effectively distinguish real signal from noisy networks. To test this, we first identified the embeddings on a simulated multilayer network using multi-node2vec. Then, we identified 13 clusters from the embeddings and calculated the adjusted rand index (as done in our application study) of the clusters with the true subnetwork labels. We repeated this across 100 simulated multilayer networks from both the healthy and patient groups.

For the simulated networks for the healthy group, we calculated an average adjusted rand of 0.25 (st. deviation 0.08). For the simulated networks representing the patient group, we calculated an average adjusted rand of 0.21 (st. dev 0.10). These results reveal that there is no structure in the embeddings of these random multilayer graph models for each group and suggest that the multi-node2vec algorithm does not incorrectly identify structure in a noisy network.

4.2.2 Sensitivity of multi-node2vec

Community Strength

We first investigate the effects of the strength of community structure on multi-node2vec. To do so, we varied the out-group probabilities p_{out} to be between 10% and 90% of p_{in} and assess the performance of the algorithm over values of the $\mathrm{SNR} = \mathrm{p_{in}}/\mathrm{p_{out}} - 1$. We simulated multilayer networks like this with 74 layers, across $c = 2$ to 14 communities per layer. Results are shown in plot (a) of Figure 2. We observe that as the disparity between in-group and out-group

probabilities increased, the feature embeddings more clearly represented the community structure in the graph. Furthermore, across all values of p_{out}, the performance of multi-node2vec improved as the number of communities decreased. For multilayer networks with two communities, the feature embeddings perfectly represented the community structure for values of SNR greater than or equal to 0.4. Networks with 14 communities per layer required SNR greater than 2.0 to achieve the same result. These results provide evidence that the feature embeddings identified by multi-node2vec are able to efficiently capture the community structure of multilayer networks.

Effect of the Number of Layers
We next analyze the effect of the number of layers on the multi-node2vec algorithm. In this simulation, we generated multilayer graphs from the planted partition model with $m = 5, 10, 15, \ldots, 65, 74$. As before, we fixed $p_{in} = 0.49$ and varied $p_{out} = 0.245, 0.196$, and 0.147 to match the best three values from the community strength simulations. We report the average adjusted rand from 30 replications on networks with $c = 12$ communities in plot (b) of Figure 2. For all three values of p_{out}, the performance of multi-node2vec consistently improves across an increasing number layers. This result supports the belief that each layer provides additional neighborhood information for each node from which the multi-node2vec algorithm can efficiently learn.

Effects of Context Size, k
To test the effect of neighborhood size, k, we ran simulations of the planted partition model multilayer networks with $m = 5, 15, 25, 50$, and74 over a range of 8–20 nodes per neighborhood with $p_{out} = 0.245$. We plot the average adjusted rand of the clusters identified on the feature matrix for networks with $c = 12$ communities in the plot (c) of Figure 2. We find that the algorithm improves with an increasing context size; however, the number of layers in the network has more impact on the performance of the algorithm. Indeed, when $m \geq 25$, the neighborhood size does not significantly affect (if at all) the performance of the algorithm. On the other hand, for a small number of layers (say, $m = 5$), the increasing the context size plays a more important role in its identified features. Thus, for multilayer networks with a large enough of layers, the context size will not dramatically affect the results of multi-node2vec, but in networks with fewer than 25 layers, one should carefully tune this parameter.

Scalability
Identifying a neighborhood for the bag of nodes needed for the algorithm relies upon a random walk strategy, which can be done in constant time using alias sampling (as done in the node2vec algorithm). The optimization part of the algorithm turns out to be linear in the number of distinct nodes in the multilayer network. Notably, this is drastically faster than the spectral decomposition of the network, which in the best-case scenario is of cubic in the unique number of nodes. To show this empirically, we consider multilayer networks with $n = 10$ unique nodes in each of m total layers. We apply multi-node2vec on planted partition networks across a range the number of layers m from 10 to 1 million layers. We calculate the amount of time (in seconds) required for multi-node2vec with fixed $k = D = 5$ on 30 replications and report the average time in the plot (d) of Figure 2. For networks with 1 million layers, multi-node2vec took on average of only 58 seconds. We note that the complexity of multi-node2vec as a function of n is also linear, and this is justified with the scalability analysis in Grover & Leskovec (2016). This figure suggests that the multi-node2vec algorithm is linear in the number of layers in the network and provides evidence that this algorithm is well suited for embedding massive multilayer networks.

4.3 Analysis of schizophrenia data

Based on our discussion and results in Sections 4.1 and 4.2, we set $k = 10$ and $l = 30$. We set $p = 1$ and $q = 0.5$ to match the parameter settings of node2vec as suggested in Grover & Leskovec (2016), and we investigated the effects of the layer walk parameter $r = 0.25, 0.50$, and 0.75.

4.3.1 Clustering ROIs

To explore functional region segmentation, we first clustered the rows of the feature matrices identified from multi-node2vec across all three walk parameter settings. For this task, we were particularly interested in the effect of the feature dimension on clustering performance. To test this effect, we proceeded as follows. Multi-node2vec was run using D features. The k-means clustering algorithm was then applied on the rows of the resulting $N \times D$ matrix, and the number of clusters was set to 13 to match the true number of subnetwork labels. For each run, the identified clusters were compared against the true subnetwork labels using the adjusted rand score. We repeated this process for each method across a grid of D from 2 to 100 in increments of 2.

The match of the identified clusters with the ground truth improves as the number of features, D increases. Notably, even for D as small as 6, the ROI clusters closely resemble the ground-truth labels (adjusted rand ≈ 0.83). We note that such clustering analyses provide a heuristic for assessing how many dimensions should be used to capture a desired ground truth in a multilayer network. For example, in this case, we can use even just two dimensions and still capture more than 80% of the functional organization of the healthy individuals. These results reveal that the features of multi-node2vec provide practically relevant information about the functional subnetwork to which these ROIs belong. This finding is further supported in the classification study performed next.

4.3.2 Classification of functional subnetworks

We now assess the utility of the features learned from multi-node2vec through the classification task of predicting the functional subnetwork location for each ROI in the healthy individuals. We considered the classification of the 9 subnetworks containing 10 or more ROIs, which included the *auditory, cingulo-opercular task control, default mode, frontoparietal task control, salience, sensory/somatomotor—hand, subcortical, visual*, and *dorsal attention* subnetworks. In the classification task, we tested two scenarios for network embedding methods—(i) the multilayer network representing the resting state fMRI of 74 healthy individuals alone and (ii) the multilayer network with additional noisy layers.

For each subnetwork, we trained a one-versus-all logistic regression classifier on the rows of the feature matrix for each method on 80% of the regions using D identified features. We applied the classifier to the remaining 20% of the ROIs and assessed the performance of the classifier using the area under the curve (AUC). We performed this classification on the feature matrices for each method and calculated the resulting AUC of the classifier across D ranging from 2 to 100 in increments of 2.

We compared multi-node2vec to several off-the-shelf embedding methods including node2vec, DeepWalk, and LINE. As these methods are single-layer methods, we ran them on the average weighted network of each population where layers were the same as those used for multi-node2vec. For node2vec, we set the return parameter as $p = 1$ and the in-out parameter as $q = 0.5$ to guide the neighborhood search following the suggestions of the original paper. For DeepWalk, we kept default parameters. Matching multi-node2vec, we set $k = 10$ for both node2vec and DeepWalk. For LINE, we used its default parameters: negative sampling $= 5$ and $\rho = 0.025$. To match LINE's default of 1 million training samples, we sampled $s = 3,788$ neighborhoods for each node in node2vec and DeepWalk. We ran all methods to learn D features, from $D = 2, \ldots, 100$. All experiments were performed on an AWS T2.Xlarge instance (specs: a 64-bit Linux platform with 16 GiB memory). We report the AUC for each method and each subnetwork when 20 layers of noise were added in Figure 3. Results for the non-noisy setting and the setting with 10 layers of noise are shown in the Appendix.

Our study reveals that even in the presence of noise, multilayer embeddings of the healthy individuals closely match the functional organization of the brain. Furthermore, multi-node2vec is comparable to the competing methods in the non-noisy setting, where we expect layers to be

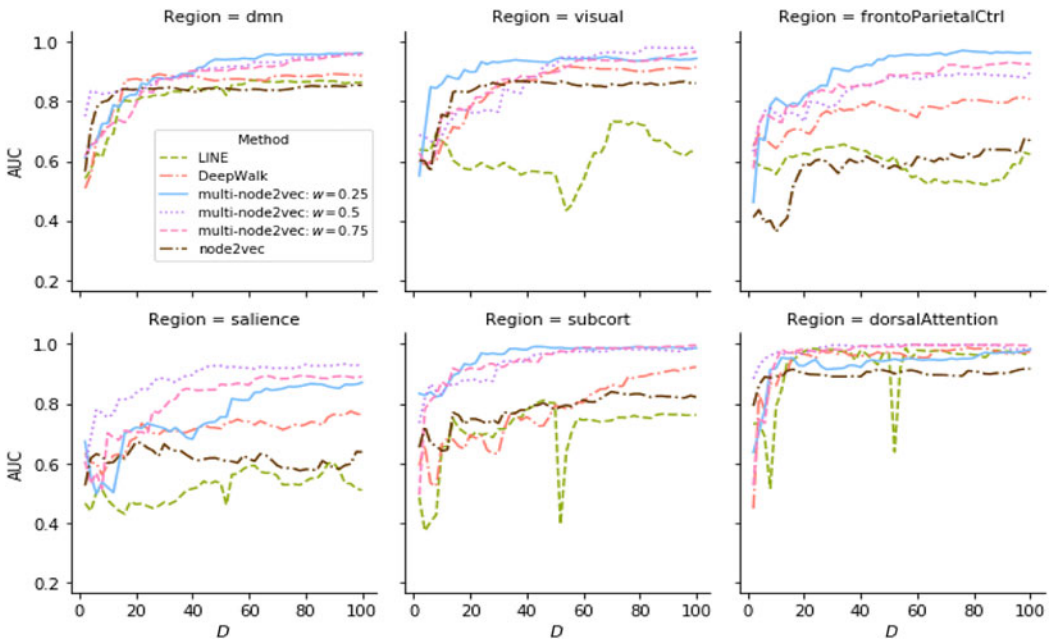

Figure 3. The AUC of a one versus all logistic regression classifier for the nine major functional subnetworks of the brain of all 74 healthy individuals and 20 layers of noise. Plots show the AUC of the classifier against the number of dimensions *D* for feature representations from multi-node2vec, node2vec, DeepWalk, and LINE.

homogeneous across the healthy patients. We further find that multi-node2vec is robust to multilayer networks with additional noisy layers. Indeed in this setting, we find that multi-node2vec outperforms its competitors in seven of nine classification studies. These results provide evidence of the robustness of multi-node2vec across multilayer networks with heterogeneous layers and reveal the overall utility of the algorithm for noisy and non-noisy networks.

We begin by analyzing the classification result on the original 74 individuals (figure shown in the Appendix.). Since each individual in the original study is healthy, we expect the networks of each these individuals to share similar structure. It follows that the aggregate network provides an unbiased summary of the multilayer network with less variability than each layer alone. Thus, methods applied to the aggregate network are expected to do better than multi-node2vec. Despite this, we find that multi-node2vec is comparable to the competing methods for seven out of nine subnetworks and outperforms other methods for small D in the *visual* and *sensory motor (hand)* regions. The LINE method does particularly well in the *salience* and *dorsal attention* classifications and outperforms multi-node2vec and all other methods across D. All methods improve with increasing D and approach 1, indicating perfect classification.

To test the performance of multi-node2vec on multilayer networks with noise, we next generated b layers, each with 264 nodes to match the number of regions in every other layer, from an Erdős–Rényi with edge probability set to the average edge density across all 74 layers. In this way, we add b layers of randomly connected nodes that act as noise against the structure present in the 74 individuals in the study. We set $b = 10$ and 20 and reran all of the methods with the same parameter settings as in the original study.

As can be seen in Figure 3, single-layer embedding methods are dramatically affected by the addition of noisy layers, whereas, multi-node2vec is robust to noise. For both $b = 10$ (in the Appendix Figure A1) and $b = 20$ (Appendix Figure A2), all three runs of multi-node2vec outperforms competing methods for seven out of nine of the classification studies. In particular, multi-node2vec has clear advantages over the competing methods in the *subcortical*, *salience*,

Table 2. Two-sided 95% confidence intervals for the difference of mean squared deviation in healthy controls and schizophrenia patients. Deviations were calculated using 100 features from the network embedding for each group

Subnetwork	$MSD_{healthy} - MSD_{patient}$	Subnetwork	$MSD_{healthy} - MSD_{patient}$
Auditory	(−0.108, 0.332)	Dorsal attention	(−0.326, 0.106)
C-O task control	(−0.319, 0.155)	Default mode	(−0.241, −0.031)†
Salience	(−0.373, 0.047)*	Memory/retrieval	(−0.359, 0.581)
F-P task control	(−0.213, 0.187)	Visual	(−0.160, 0.095)
Sensory—hand	(−0.295, 0.089)	Ventral attention	(−0.204, 0.434)
subcortical	(−0.198, 0.488)	Cerebellar	(−0.578, 0.379)
Sensory—mouth	(−0.440, 0.269)		

*The sum of squares deviation in healthy controls was less than that in the schizophrenia patients at a 0.10 level.
†The sum of squares deviation in healthy controls was less than that in the patients with schizophrenia at a 0.001 level

sensory motor (hand), and *frontoparietal task control* regions. Importantly, multi-node2vec's performance is not strongly affected by the addition of more noisy layers suggesting that the features identified by the method align with the true 74 layers of the population. We find that the LINE method is most affected by noise, followed by node2vec.

These results, in combination to the clustering results from the previous section, provide strong evidence that the features engineered from multi-node2vec provide biologically relevant information about the functional organization of the brain and is generally robust to moderate amounts of noisy layers.

4.3.3 Comparison of healthy controls and patients

To compare the populations of patients with schizophrenia to their healthy peers, we apply multi-node2vec with $r = 0.25$ to both groups using the same parameters as described above providing a 264×100 network embedding for each group. Importantly, the two embeddings are not directly comparable as features for each population may differ or be arranged in differing order. To assess differences between groups, then one must compare within population summaries across populations. For our study, we compare the variability of the embeddings within each functional subnetwork. To make this precise, let A represent the index of the regions that are contained within a specified functional subnetwork $\mathcal{A}$. Let $f_{g,i}$ denote the ith feature vector in group g and $\bar{f}_{g,\mathcal{A}}$ denote the mean vector of the embeddings from region $\mathcal{A}$ in group g, where $g =$healthy patient. Let $||\, x\, ||_F$ notate the Frobenius norm of the vector x. For each group, we calculate the mean squared deviation for every region $\mathcal{A}$:

$$MSD_{g,\mathcal{A}} = \frac{1}{|\mathcal{A}|} \sum_{i \in A} ||\, f_{g,i} - \bar{f}_{g,\mathcal{A}}\, ||_F^2$$

where $|\cdot|$ represents the cardinality of a set. The value of $MSD_{g,\mathcal{A}}$ quantifies the inner regional variability of the embeddings for region $\mathcal{A}$ in the gth sample. Large values of $MSD_{g,\mathcal{A}}$ suggest low similarity of nodes within the same region $\mathcal{A}$ and hence higher entropy among that region. For each region $\mathcal{A}$ mentioned in Table 1, we compare the mean squared deviation across populations using a two-sided t-test on the quantity:

$$MSD_{healthy,\mathcal{A}} - MSD_{patient,\mathcal{A}}$$

These results are reported in Table 2. We find a significant difference in the mean squared deviation in the default mode network (DMN) (p-value < 0.001) as well as a strong trend within the salience network (p-value = 0.085). In both subregions, the mean squared deviation was found to be lower in the healthy group than in the patient population, suggesting higher variability in the

patient group in these two regions. Our findings are well supported by the triple network model (TNM) theory of the brain (Menon & Uddin, 2010; Seeley et al., 2007). The TNM explains how individuals switch between externally motivated cognitive processes (i.e., goal-directed tasks) that are associated with the central executive networks and internally motivated cognitive processes (i.e., rumination and mind-wandering) that are associated with DMN via the salience network (Menon & Uddin, 2010; Seeley et al., 2007).

The TNM foremost relates to schizophrenia because of differences observed in the DMN in individuals with schizophrenia. Previous research has indicated increased activity and within-network connectivity in the DMN (Whitfield-Gabrieli et al., 2009) and decreased segregation between the DMN and central executive networks (Woodward et al., 2011) in patients with schizophrenia versus healthy individuals. The TNM further posits that pathological salience (inappropriate monitoring by the salience network) may be associated with DMN pathology and consequently many of the symptoms of schizophrenia (Menon, 2011). This theory is consistent with recent evidence indicating TNM, and particularly salience network, dysregulation is correlated with symptom severity in patients with schizophrenia (Hare et al., 2018; Supekar et al., 2019). Our findings that the DMN has significantly smaller variability within healthy individuals than in individuals with schizophrenia as well as the fact that the salience network is statistically different between individuals with schizophrenia and healthy controls empirically supports these findings.

Finally, our results are consistent with a recent meta-analysis investigating the effect of schizophrenia on connectivity (Li et al., 2019), which found consistent hypoconnectivity among the DMN in patients with schizophrenia. Notably, however, this meta-analysis also found aberrant connections in several other functional networks, a finding we do not replicate here. Future research is clearly needed to know whether our lack of significant findings in other networks (e.g., auditory, somatomotor) reflects lower power in our study compared to the meta-analysis, or a systematic difference as a result of the vastly different methodological approaches. Given that our results with the DMN and the salience network are consistent with both the meta-analysis as well as other papers using this same dataset (e.g., Wang et al., 2014), we suspect this is primarily an issue of power, but future work is clearly necessary to fully understand these discrepancies.

4.3.4 Classification of patients and healthy controls

We next consider the classification task of differentiating schizophrenia patients from healthy controls using the embeddings from multi-node2vec. We first apply multi-node2vec to each of the 134 total individuals in the study separately (74 healthy and 60 patients) and extract $D = 100$ feature embeddings describing each person's functional connectivity. From these embeddings, we then calculate the mean squared deviance $MSD_{j,\mathcal{A}}$ for each individual $j = 1, \ldots, 134$ and each region $\mathcal{A}$. Using the binary response vector $y = (y_1, \ldots, y_{134})$ where $y_j = 1$ if individual j has schizophrenia and 0 if individual j is a healthy control, we apply several off-the-shelf binary classification techniques—including k-nearest neighbors, logistic regression, an L2 penalized logistic regression, and a random forest classifier—using the mean squared deviance vectors to predict whether or not the individual has schizophrenia. We perform 10-fold cross-validation and report the average and standard error of the results in Table 3. For k-nearest neighbors, we look across a grid of k between 1 and 30 and report the result with the highest mean accuracy.

The random forest classifier performs better than the other off-the-shelf methods using our discovered embeddings and obtains a classification accuracy of 0.787 on average. It is important to reiterate that multi-node2vec is an *unsupervised method*, namely the algorithm is not trained to explicitly distinguish between two populations as is done formally in the network classification problem. With that in mind, there have studies on the COBRE dataset that were *supervised* and though these studies are not directly comparable with our result, their comparison does deserve some discussion.

We compare our findings with the recent work in Relión et al. (2019), which establishes the highest performance to date on the COBRE dataset using supervised edge-based techniques (see

Table 3. Ten-fold cross-validation results for classification of patients and healthy controls using individual embeddings. The average and standard error (s.e.) are reported

Method	Mean accuracy (s.e.)
k-nearest neighbors	0.718 (0.017)
Logistic regression	0.758 (0.075)
L2 penalized logistic	0.592 (0.021)
Random forest	0.787 (0.077)

Table 1 for their results). Their method achieved an accuracy of 0.927. Furthermore, other edge-based methods that employ variable selection on the edges in each network obtain accuracies on average of approximately 0.85. As expected, such supervised methods do indeed outperform our unsupervised strategy. Perhaps the most fair comparison among the results in Relión et al. (2019) to our own result is the comparison of multi-node2vec with network summaries. Like multi-node2vec, network summaries provide a dimension reduction to the original networks and are not explicitly designed for classification. We find that classification via embeddings of multi-node2vec significantly outperform the classification using network summaries, which obtained 0.614 accuracy on average. This study reinforces the fact that multi-node2vec provides biologically relevant information for classification of disease type. In future work, we will investigate developing supervised embedding methods designed specifically to classify disease and other clinical features.

5. Limiting behavior of multi-node2vec

Multi-node2vec is an approximate algorithm that seeks to maximize the log-likelihood objective function given in Equation (5). Approximation is needed for two objectives—(i) the identification of multilayer neighborhoods via random walks and (ii) the application of the Skip-gram neural network model with negative sampling. By analyzing the asymptotic nature of the random walks in the NeighborhoodSearch procedure as $l \to \infty$, one can leverage the recent work on the Skip-gram with negative sampling from Levy & Goldberg (2014), Qiu et al. (2018) to show that multi-node2vec approximates implicit matrix factorization. We describe this main result below.

Denote $\mathcal{D}$ as the collection of neighborhoods identified by the NeighborhoodSearch procedure. Let $w = \{u_1, \ldots, u_l\} \in \mathcal{D}$ be a collection of nodes resulting from a length l random walk in the NeighborhoodSearch procedure. Define the k-length contexts for node u_i as the nodes neighborhing it in a k-sized window $u_{i-k}, \ldots, u_{i-1}, u_{i+1}, \ldots, u_{i+k}$ and let c denote the collection of contexts for w. Let $\#(w, c)$ denote the number of times the node–context pair (w, c) appears in $\mathcal{D}$. Further, let $\#(w)$ and $\#(c)$ denote the number of times the node w and the context c appear in $\mathcal{D}$, respectively. As shown in Levy & Goldberg (2014), running Skip-gram with negative sampling is equivalent to implicitly factorizing:

$$\log\left(\frac{\#(w,c)|\mathcal{D}|}{\#(w)\#(c)}\right) - \log(b) \tag{9}$$

where b is the number of negative samples specified. Expression (9) suggests that by getting a hold of the quantity in the first logarithm of the expression, we can relate multi-node2vec directly to matrix factorization.

Our results provide asymptotic expressions for $\#(w, c)|\mathcal{D}|/\#(w)\#(c)$ when the random walk length $l \to \infty$. To make our result explicit, we need to first introduce a little more notation. Define $\widetilde{\mathbf{d}}_u = \sum_{v \in \mathcal{N}} \widetilde{A}_{u,v}(r)$ as the generalized degree of node u in $\widetilde{\mathbf{G}}_{\mathcal{N}}(r)$ and let $\widetilde{\mathbf{D}} = \mathrm{diag}(\widetilde{\mathbf{d}}_1, \ldots, \widetilde{\mathbf{d}}_N)$. Define the volume of $\mathbf{G}_{\mathcal{N}}(r)$ as $\mathrm{vol}(\widetilde{\mathbf{G}}_{\mathcal{N}}(r)) = \sum_{u \in \mathcal{N}} \widetilde{\mathbf{d}}_u$. Define $\underline{\mathbf{P}}$ as the array containing

the second-order transition probabilities of NeighborhoodSearch: $\underline{\mathbf{P}} = \{\underline{P}_{u,v,w} = P(u_{j+1} = u \mid u_j = v, u_{j-1} = w)\}$ and let $\mathbf{X}$ be its corresponding stationary distribution satisfying $\sum_w \underline{P}_{u,v,w} X_{v,w} = X_{u,v}$. Furthermore, let $\underline{P}^k_{u,v,w} = P(u_{j+r} = u \mid u_j = v, u_{j-1} = w)\}$ denote the kth step transition probability.

Finally, suppose $\xrightarrow{P}$ denotes convergence in probability. Our analysis of multi-node2vec depend on the bias of the transition probabilities for the random walks of the NeighborhoodSearch procedure in Equation (7), $\alpha_{pqr}(t, x, \ell, \ell')$. We can now state our next theorem, which relates multi-node2vec directly with matrix factorization.

Theorem 2. *Let $\mathbf{G}^m_{\mathcal{N}}$ be an observed multilayer network and let $\widetilde{\mathbf{G}}_{\mathcal{N}}(r)$ be its adjusted aggregate network. Suppose that $\widetilde{\mathbf{G}}_{\mathcal{N}}(r)$ is connected, undirected, and non-bipartite. Let k be the context size chosen for the Optimization procedure. Then as $l \to \infty$,*

(a) *For all $p, q, r > 0$,*

$$\frac{\#(w,c)|\mathcal{D}|}{\#(w)\#(c)} \xrightarrow{P} \frac{1}{2k} \frac{\sum_{j=1}^{k} \left(\sum_u X_{w,u} \underline{P}^j_{c,w,u} + \sum_u X_{c,u} \underline{P}^j_{w,c,u} \right)}{\left(\sum_u X_{w,u} \right) \left(\sum_u X_{c,u} \right)} \tag{10}$$

(b) *Let $\widetilde{\mathbf{P}} = \widetilde{\mathbf{D}}^{-1} \widetilde{\mathbb{A}}$. If $p = q = 1$,*

$$\frac{\#(w,c)|\mathcal{D}|}{\#(w)\#(c)} \xrightarrow{P} \frac{vol(\widetilde{\mathbf{G}}_{\mathcal{N}}(r))}{k} \left(\sum_{x=1}^{k} \widetilde{\mathbf{P}}^k \right) \widetilde{\mathbf{D}}^{-1} \tag{11}$$

for all $r > 0$.

By applying the result of Lemma 3.1, we can apply Theorems 2.1–2.3 and result (8) from Qiu et al. (2018) directly to prove the Theorem 3. Results (10) and (11) provide closed-form limiting expressions for the matrix factorization problem in Equation (9). These results suggest the use of matrix factorization to identify features for a multilayer network; however, it should be noted that calculating and storing the second-order transition probabilities $\underline{\mathbf{P}}$ and its stationary distribution $\mathbf{X}$ is computationally prohibitive. We do not consider such an algorithm in our current study but plan to address fast matrix factorization in future work.

6. Discussion

In this paper, we introduced the multi-node2vec algorithm, the fast network embedding technique for complex multilayer networks. This work motivates several areas of future work. For example, an important next step is to incorporate partial supervision for the detection of relevant features that depend on the application under investigation. Recent work like Kipf & Welling (2016) for semi-supervised feature engineering on static networks may provide a principled first step in the investigation for multilayer networks. We furthermore believe that it will be fruitful to thoroughly compare and contrast feature engineering methods like multi-node2vec with the results of multilayer community detection methods so as to better understand the discovered features. Furthermore, though not explicitly considered here, multi-node2vec is readily applicable to dynamic networks, an example of multilayer networks where the ordering of layers depends on time. This work will require incorporating appropriate notions of conditional dependence between the layers that replace the conditional independence assumptions applied here. Finally, our theoretical analysis of the multi-node2vec algorithm motivates further work in understanding the relationship between neural network algorithms with more traditional machine learning tasks such as matrix factorization. We believe that more work should be done in this area to fully understand the theoretical underpinnings of deep learning.

There are several aspects of the multi-node2vec algorithm that open up new directions of research. First, there is a need for an embedding procedure for dynamic networks that incorporates dependencies between each observed network in a sequence. This is particularly of interest for functional connectivity data observed through time, like the raw data considered in this paper. With multi-node2vec, a dynamic generalization is possible through the relaxing of simplifying assumptions (A1) and (A2) to allow for dependence between neighborhoods across time. Alternatively, one could directly construct an embedding algorithm for dynamic generative models like the family of temporal latent space models (Sewell & Chen, 2015) or temporal exponential random graph models (Hanneke et al., 2010; Lee et al., 2020). Second, in this paper, we treated a population as a collection of people across scans. However, in many instances, multiple scans like different tasks, for example, are available for each individual. It would be interesting and perhaps very fruitful to obtain a multilayer embedding for each individual instead using a multilayer network collected across all the available scans. Finally, much work has recently focused on mixture models for populations of networks, where the networks themselves cluster. One could account for such structure in multi-node2vec by again reworking the assumptions in (A1) and (A2) to account for differences between clusters of networks.

The multi-node2vec technique has potential for ground-breaking discovery in the study of functional connectivity. By specifying a multilayered framework that (i) models weighted networks, (ii) does not require temporal ordering of the layers, and (iii) is robust to noisy layers, multi-node2vec enables the study of networks that vary across individuals and cognitive tasks. Neuroscientists have very recently begun to utilize multilayer analyses (Betzel & Bassett, 2017; Bassett et al., 2015, 2011; Muldoon & Bassett, 2016; Braun et al., 2015). The majority of this work has explored how community structure and network modules vary across time. For instance, one study showed that shifts in community structure across time predict differences in learning a visual motor task (Bassett et al., 2015). Indeed, network neuroscientists have lately called for greater emphasis on multilayer techniques, particularly those that do not require temporal ordering of layers, thus allowing for more comprehensive quantification of networks across samples (Muldoon & Bassett, 2016). The multi-node2vec algorithm is a fully data-driven strategy with the capabilities to learn significant neurological variation among brains and will progress the investigation of individual differences and disease.

Funding. JDW gratefully acknowledges support on this project by the National Science Foundation grant NSF DMS-1830547.

Conflicts of interest. None.

References

Achard, S., Salvador, R., Whitcher, B., Suckling, J., & Bullmore, E. D. (2006). A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *The Journal of Neuroscience*, *26*(1), 63–72.

Bassett, D. S., & Bullmore, E. D. (2006). Small-world brain networks. *The Neuroscientist*, *12*(6), 512–523.

Bassett, D. S., Meyer-Lindenberg, A., Achard, S., Duke, T., & Bullmore, E. (2006). Adaptive reconfiguration of fractal small-world human brain functional networks. *Proceedings of the National Academy of Sciences*, *103*(51), 19518–19523.

Bassett, D. S., Wymbs, N. F., Porter, M. A., Mucha, P. J., Carlson, J. M., & Grafton, S. T. (2011). Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, *108*(18), 7641–7646.

Bassett, D. S., Yang, M., Wymbs, N. F., & Grafton, S. T. (2015). Learning-induced autonomy of sensorimotor systems. *Nature Neuroscience*, *18*(5), 744–751.

Betzel, R. F., & Bassett, D. S. (2017). Multi-scale brain networks. *Neuroimage*, *160*, 73–83.

Biswal, B. B., Mennes, M., Zuo, X.-N., Gohel, S., Kelly, C., Smith, S. M., ... Milham, M. P. (2010). Toward discovery science of human brain function. *Proceedings of the National Academy of Sciences*, *107*(10), 4734–4739.

Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C. I., Gómez-Gardeñes, J., Romance, M., ... Zanin, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, *544*(1), 1–122.

Braun, U., Schäfer, A., Walter, H., Erk, S., Romanczuk-Seiferth, N., Haddad, L., ... Bassett, D. S. (2015). Dynamic reconfiguration of frontal brain networks during executive cognition in humans. *Proceedings of the National Academy of Sciences*, *112*(37), 11678–11683.

Bressler, S. L., & Menon, V. (2010). Large-scale brain networks in cognition: Emerging methods and principles. *Trends in Cognitive Sciences*, *14*(6), 277–290.
Bühlmann, P., & Van De Geer, S. (2011). *Statistics for high-dimensional data: Methods, theory and applications*. Springer-Verlag Berlin Heidelberg London New York: Springer Science & Business Media.
Bullmore, Ed., & Sporns, O. (2009). Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, *10*(3), 186–198.
Bullmore, Ed., & Sporns, O. (2012). The economy of brain network organization. *Nature Reviews Neuroscience*, *13*(5), 336–349.
Chai, X. J., Castañón, A. N., Öngür, D., & Whitfield-Gabrieli, S. (2012). Anticorrelations in resting state networks without global signal regression. *Neuroimage*, *59*(2), 1420–1428.
Cole, M. W., Yarkoni, T., Repovš, G., Anticevic, A., & Braver, T. S. (2012). Global connectivity of prefrontal cortex predicts cognitive control and intelligence. *The Journal of Neuroscience*, *32*(26), 8988–8999.
De Domenico, M., Lancichinetti, A., Arenas, A., & Rosvall, M. (2015). Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X*, *5*(1), 011027.
Denny, M. J., Wilson, J. D., Cranmer, S. J., Desmarais, B. A., & Bhamidi, S. (2017). Gergm: Estimation and fit diagnostics for generalized exponential random graph models. *R package version 0.11*, *2*.
Fornito, A., Zalesky, A., Bassett, D. S., Meunier, D., Ellison-Wright, I., Yücel, M., ... Bullmore, E. T. (2011). Genetic influences on cost-efficient organization of human cortical functional networks. *The Journal of Neuroscience*, *31*(9), 3261–3270.
Gallagher, B., & Eliassi-Rad, T. (2010). Leveraging label-independent features for classification in sparsely labeled networks: An empirical study. In *Advances in social network mining and analysis* (pp. 1–19). Springer.
Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, *151*, 78–94.
Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864). ACM.
Han, Q., Xu, K., & Airoldi, E. (2015). Consistent estimation of dynamic and multi-layer block models. In *Proceedings of the 32nd international conference on machine learning (ICML-15)* (pp. 1511–1520).
Hanneke, S., Fu, W., & Xing, E. P. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics*, *4*, 585–605.
Hare, S. M., Ford, J. M., Mathalon, D. H., Damaraju, E., Bustillo, J., Belger, A., ... Turner, J. A. (2018). Salience–default mode functional network connectivity linked to positive and negative symptoms of schizophrenia. *Schizophrenia Bulletin*, *45*(4), 892–901.
He, Y., Chen, Z. J., & Evans, A. C. (2007). Small-world anatomical networks in the human brain revealed by cortical thickness from MRI. *Cerebral Cortex*, *17*(10), 2407–2419.
Henderson, K., Gallagher, B., Li, L., Akoglu, L., Eliassi-Rad, T., Tong, H., & Faloutsos, C. (2011). It's who you know: Graph mining using recursive structural features. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 663–671). ACM.
Hoff, P. D., Raftery, A. E., & Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, *97*(460), 1090–1098.
Insel, T. R., Landis, S. C., & Collins, F. S. (2013). The NIH brain initiative. *Science*, *340*(6133), 687–688.
Kinnison, J., Padmala, S., Choi, J.-M., & Pessoa, L. (2012). Network analysis reveals increased integration during emotional and motivational processing. *The Journal of Neuroscience*, *32*(24), 8361–8372.
Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arxiv preprint arxiv:1609.02907.
Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., & Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, *2*(3), 203–271.
Lambiotte, R., Delvenne, J.-C., & Barahona, M. (2008). Laplacian dynamics and multiscale modular structure in networks. arxiv preprint arxiv:0812.1770.
Lee, J. D., Simchowitz, M., Jordan, M. I., & Recht, B. (2016). Gradient descent only converges to minimizers. In *Conference on learning theory* (pp. 1246–1257).
Lee, J., Li, G., & Wilson, J. D. (2020). Varying-coefficient models for dynamic networks. *Computational Statistics & Data Analysis*, *152*, 107052.
Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems* (pp. 2177–2185).
Li, S., Hu, N., Zhang, W., Tao, B., Dai, J., Gong, Y., ... Lui, S. (2019). Dysconnectivity of multiple brain networks in schizophrenia: A meta-analysis of resting-state functional connectivity. *Frontiers in Psychiatry*, *10*, 482.
Mayer, A. R., Ruhl, D., Merideth, F., Ling, J., Hanlon, F. M., Bustillo, J., & Cañive, J. (2013). Functional imaging of the hemodynamic sensory gating response in schizophrenia. *Human Brain Mapping*, *34*(9), 2302–2312.
McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, *27*(1), 415–444.

Medaglia, J. D., Lynall, M.-E., & Bassett, D. S. (2015). Cognitive network neuroscience. *Journal of Cognitive Neuroscience*, *27*(8), 1471–1491.
Menon, V. (2011). Large-scale brain networks and psychopathology: A unifying triple network model. *Trends in Cognitive Sciences*, *15*(10), 483–506.
Menon, V., & Uddin, L. Q. (2010). Saliency, switching, attention and control: A network model of insula function. *Brain Structure and Function*, *214*(5–6), 655–667.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient estimation of word representations in vector space. arxiv preprint arxiv:1301.3781.
Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., & Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *Science*, *328*(5980), 876–878.
Muldoon, S. F., & Bassett, D. S. (2016). Network and multilayer network approaches to understanding human brain dynamics. *Philosophy of Science*, *83*(5), 710–720.
Paul, S., & Chen, Y. (2018). A random effects stochastic block model for joint community detection in multiple networks with applications to neuroimaging. arxiv preprint arxiv:1805.02292.
Pavlovic, D. M., Guillaume, B. L. R., Towlson, E. K., Kuek, N. M. Y., Afyouni, S., Vertes, P. E., ... Nichols, T. E. (2019). Multi-subject stochastic blockmodels for adaptive analysis of individual differences in human brain network cluster structure. Biorxiv, 672071.
Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP* (pp. 1532–1543), vol. 14.
Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 701–710). ACM.
Power, J. D., Cohen, A. L., Nelson, S. M., Wig, G. S., Barnes, K. A., Church, J. A., ... Petersen, S. E. (2011). Functional network organization of the human brain. *Neuron*, *72*(4), 665–678.
Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., & Tang, J. (2018). Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 459–467). ACM.
Relión, J. D. A., Kessler, D., Levina, E., & Taylor, S. F. (2019). Network classification with applications to brain connectomics. *The Annals of Applied Statistics*, *13*(3), 1648–1677.
Rosenthal, G., Váša, F., Griffa, A., Hagmann, P., Amico, E., Goñi, J., ... Sporns, O. (2018). Mapping higher-order relations between brain structure and function with embedded vector representations of connectomes. *Nature Communications*, *9*(1), 2178.
Seeley, W. W., Menon, V., Schatzberg, A. F., Keller, J., Glover, G. H., Kenna, H., ... Greicius, M. D. (2007). Dissociable intrinsic connectivity networks for salience processing and executive control. *The Journal of Neuroscience*, *27*(9), 2349–2356.
Sewell, D. K., & Chen, Y. (2015). Latent space models for dynamic networks. *Journal of the American Statistical Association*, *110*(512), 1646–1657.
Simpson, S. L., Bahrami, M., & Laurienti, P. J. (2019). A mixed-modeling framework for analyzing multitask whole-brain network data. *Network Neuroscience*, *3*(2), 307–324.
Smith, S. M., Miller, K. L., Salimi-Khorshidi, G., Webster, M., Beckmann, C. F., Nichols, T. E., ... Woolrich, M. W. (2011). Network modelling methods for FMRI. *Neuroimage*, *54*(2), 875–891.
Sporns, O. (2011). *Networks of the brain*. MIT press.
Sporns, O. (2014). Contributions and challenges for network models in cognitive neuroscience. *Nature Neuroscience*, *17*(5), 652–660.
Sporns, O., & Betzel, R. F. (2016). Modular brain networks. *Annual Review of Psychology*, *67*, 613.
Stanley, N., Shai, S., Taylor, D., & Mucha, P. (2016). Clustering network layers with the strata multilayer stochastic block model. *IEEE transactions on network science and engineering*, *3*(2), 95–105.
Stillman, P. E., Wilson, J. D., Denny, M. J., Desmarais, B. A., Bhamidi, S., Cranmer, S. J., & Lu, Z.-L. (2017). Statistical modeling of the default mode brain network reveals a segregated highway structure. *Scientific Reports*, *7*(1), 1–14.
Stillman, P. E., Wilson, J. D., Denny, M. J., Desmarais, B. A., Cranmer, S. J., & Lu, Z.-L. (2019). A consistent organizational structure across multiple functional subnetworks of the human brain. *Neuroimage*, *197*, 24–36.
Supekar, K., Cai, W., Krishnadas, R., Palaniyappan, L., & Menon, V. (2019). Dysregulated brain dynamics in a triple-network saliency model of schizophrenia and its relation to psychosis. *Biological Psychiatry*, *85*(1), 60–69.
Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web* (pp. 1067–1077). International World Wide Web Conferences Steering Committee.
van den Heuvel, M. P., Kahn, R. S., Goñi, J., & Sporns, O. (2012). High-cost, high-capacity backbone for global brain communication. *Proceedings of the National Academy of Sciences*, *109*(28), 11372–11377.
van den Heuvel, M. P., & Sporns, O. (2011). Rich-club organization of the human connectome. *The Journal of Neuroscience*, *31*(44), 15775–15786.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E. J., Yacoub, E., Ugurbil, K., & for the WU-Minn HCP Consortium. (2013). The WU-Minn Human connectome project: An overview. *Neuroimage*, *80*, 62–79.
Van Essen, D. C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T. E. J., Bucholz, R., ... WU-Minn HCP Consortium. (2012). The human connectome project: A data acquisition perspective. *Neuroimage*, *62*(4), 2222–2231.
Wang, L., Zou, F., Shao, Y., Ye, E., Jin, X., Tan, S., ... Yang, Z. (2014). Disruptive changes of cerebellar functional connectivity with the default mode network in schizophrenia. *Schizophrenia Research*, *160*(1–3), 67–72.
Whitfield-Gabrieli, S., Thermenos, H. W., Milanovic, S., Tsuang, M. T., Faraone, S. V., McCarley, R. W., ... Seidman, L. J. (2009). Hyperactivity and hyperconnectivity of the default network in schizophrenia and in first-degree relatives of persons with schizophrenia. *Proceedings of the National Academy of Sciences*, pnas–0809141106.
Wilson, J. D., Cranmer, S., & Lu, Z.-L. (2020). A hierarchical latent space network model for population studies of functional connectivity. *Computational Brain & Behavior*, 1–16. doi: 10.1007/s42113-020-00080-0
Wilson, J. D., Palowitch, J., Bhamidi, S., & Nobel, A. B. (2017a). Community extraction in multilayer networks with heterogeneous community structure. *The Journal of Machine Learning Research*, *18*(1), 5458–5506.
Wilson, J. D., Denny, M. J., Bhamidi, S., Cranmer, S. J., & Desmarais, B. A. (2017b). Stochastic weighted graphs: Flexible model specification and simulation. *Social Networks*, *49*, 37–47.
Woodward, N. D., Rogers, B., & Heckers, S. (2011). Functional resting-state networks are differentially affected in schizophrenia. *Schizophrenia Research*, *130*(1–3), 86–93.

Appendix: Additional results for subnetwork classification study

Below, we provide the AUC across competing methods for classification of functional subnetworks in healthy individuals. These results illustrate the results when the methods are applied across the population of healthy individuals (Figure A1) and when applied to the population of healthy individuals with 10 layers of noise added to the network (Figure A2). These results complement those already provided and discussed in Section 4.3.

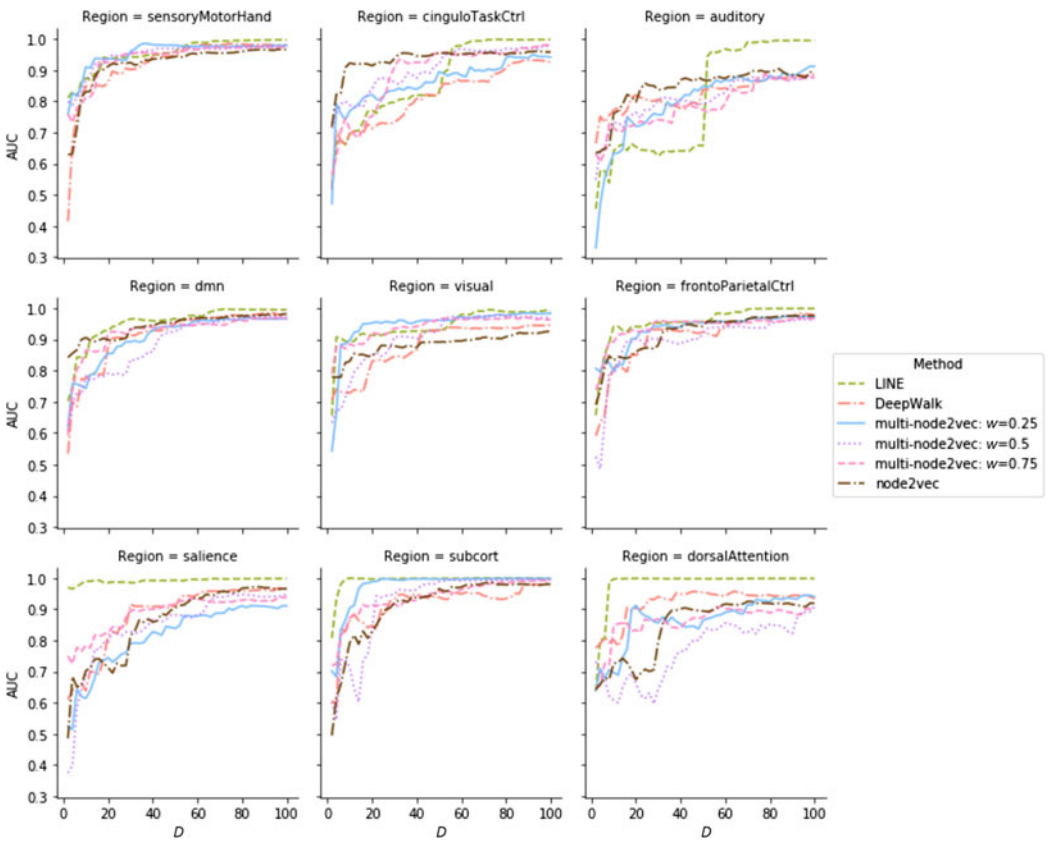

Figure A1. The AUC of a one versus all logistic regression classifier for the 9 major functional subnetworks of the brain across 74 healthy individuals. Plots show the AUC of the classifier against the number of dimensions *D* for feature representations from multi-node2vec, node2vec, DeepWalk, LINE, and the spectral decomposition.

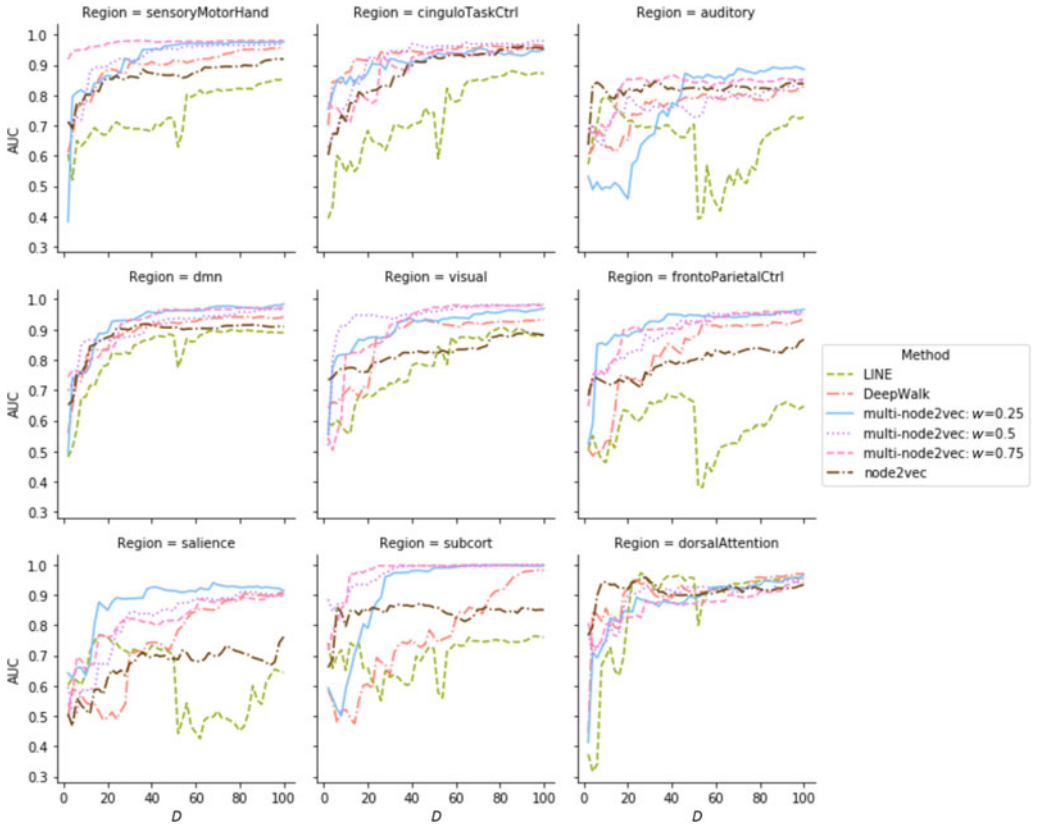

Figure A2. The AUC of a one versus all logistic regression classifier for the 9 major functional subnetworks of the brain across all 74 healthy individuals and 10 layers of noise. Plots show the AUC of the classifier against the number of dimensions D for feature representations from multi-node2vec, node2vec, DeepWalk, LINE, and spectral decomposition.

Cite this article: Wilson J. D., Baybay M., Sankar R., Stillman P. and Popa A. M. (2021). Analysis of population functional connectivity data via multilayer network embeddings. *Network Science* **9**, 99–122. https://doi.org/10.1017/nws.2020.39