
NeuroEvolve: A Dynamic Brain Graph Deep Generative Model

Alexander Campbell^{*12} Simeon Spasov^{*13} Nicola Toschi⁴⁵ Pietro Liò¹

Abstract

Graphs are a natural representation of brain activity derived from functional magnetic imaging (fMRI) data. It is well known that communities of nodes extracted from brain graphs, referred to as functional connectivity networks (FCNs), serve as useful biomarkers for understanding brain function and dysfunction. Previous works, however, ignore the temporal dynamics of the brain and focus on static graph representations. In this paper we propose NeuroEvolve, a dynamic brain graph deep generative model which simultaneously learns graph-, node-, and community-level embeddings in an unsupervised fashion. Specifically, NeuroEvolve represents brain graph nodes as embeddings sampled from a distribution over communities that evolve over time. The community distribution is parameterized using neural networks that learn from subject and node embeddings as well as past community assignments. Experiments on real-world fMRI data demonstrate NeuroEvolve outperforms state-of-the-art baselines in graph generation, dynamic link prediction, and is comparable for graph classification. Finally, an interpretability analysis of the learnt community distributions reveals overlap with known FCNs reported in neuroscience literature.

1. Introduction

Functional magnetic resonance imaging (fMRI) is a non-invasive imaging technique primarily used for measur-

ing blood-oxygen level dependent (BOLD) signal in the brain (Huettel et al., 2004). A natural representation of fMRI data is as a discrete-time graph, henceforth referred to as a dynamic brain graph (DBG), consisting of a set of fixed nodes corresponding to anatomically separated brain regions (Lawrence et al., 2021; Hess et al., 2018) and a set of time-varying edges determined by a measure of dynamic functional connectivity (FC) (Calhoun et al., 2014; Hutchison et al., 2013). DBGs have been widely used in graph-based network analysis for understanding brain function (Hirsch & Wohlschlaeger, 2022; Raz et al., 2016) and dysfunction (Alonso Martínez et al., 2020; Dautricourt et al., 2022; Yu et al., 2015).

Recently, there is growing interest in using deep learning-based methods for learning representations of graph-structured data (Goyal & Ferrara, 2018; Hamilton, 2020). A graph representation typically consists of a low-dimensional vector embedding of either the entire graph (Narayanan et al., 2017) or a part of its structure such as nodes (Grover & Leskovec, 2016), edges (Gao et al., 2019), or subgraphs (Adhikari et al., 2017). Although originally formulated for static graphs, several existing methods have been extended (Mahdavi et al., 2018; Goyal et al., 2020), and new ones proposed (Zhou et al., 2018; Sankar et al., 2020), for dynamic graphs. The embeddings are usually learnt in either a supervised or unsupervised fashion and are typically used in tasks such as node classification (Pareja et al., 2020) and dynamic link prediction (Goyal et al., 2018).

To date, very few deep learning-based methods have been designed for, or existing methods applied to, representation learning of DBGs. Those that do tend to use graph neural networks (GNNs) (Wu et al., 2020) that are designed for learning node- and graph-level embeddings (Kim et al., 2021; Dahan et al., 2021). Although these embeddings are effective at representing local/global graph structure, they are less adept at representing topological structures in-between these two extremes such as clusters of nodes or communities (Wang et al., 2017). Recent methods that explicitly incorporate community embeddings alongside node embeddings have shown improved performance for static graph representation learning tasks (Sun et al., 2019; Cavallari et al., 2017). How to leverage the relatedness of graph, node, and community embeddings in a unified framework for DBG representation learning remains under-explored.

^{*}Equal contribution ¹Department of Computer Science and Technology, University of Cambridge, Cambridge, United Kingdom ²The Alan Turing Institute, London, United Kingdom ³German Center for Neurodegenerative Diseases, Bonn, Germany ⁴University of Rome Tor Vergata, Rome, Italy ⁵A.A. Martinos Center for Biomedical Imaging, Harvard Medical School, Boston, United States. Correspondence to: Alexander Campbell <ajrc4@cl.cam.ac.uk>, Simeon Spasov <ses88@cl.cam.ac.uk>.

Workshop on Interpretable ML in Healthcare at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

Contributions To address these shortcomings, we propose NeuroEvolve¹, a hierarchical structured deep generative model (DGM) specifically designed for unsupervised learning of DBGs derived from multi-subject fMRI data. NeuroEvolve combines graph, node, and community embeddings in a unified framework, utilizing neural networks (NNs) to parameterize a community distribution over the nodes that evolves over time. NeuroEvolve also incorporates inductive biases in its structure inspired from prior knowledge about brain FCNs. We evaluate NeuroEvolve on multiple real-world fMRI datasets and show that it outperforms state-of-the-art baselines for graph reconstruction, dynamic link prediction, and achieves comparable results for graph classification.

2. Related work

Dynamic graph generative models Classic generative models for graph-structured data are typically designed for modeling a small set of specific properties (e.g. degree distribution, eigenvalues, modularity) of static graphs (Erdos et al., 1960; Barabási & Albert, 1999; Nowicki & Sniijders, 2001). DGMs that exploit the learning capacity of NNs are able to learn more expressive graph distributions (Mehta et al., 2019; Kipf & Welling, 2016b; Sarkar et al., 2020). Recent DGMs for dynamic graphs are majority VAE-based (Kingma & Welling, 2013) and are unable to learn community representations (Hajiramezanali et al., 2019; Gracious et al., 2021; Zhang et al., 2021). The few that do, are designed for static graphs (Sun et al., 2019; Khan et al., 2021; Cavallari et al., 2017).

Learning representations of dynamic brain graphs BOLD signals derived from fMRI, whether at the voxel or brain region level, represent non stationary timeseries (Guan et al., 2020). As such, how BOLD signals relate to each other spatially changes over time. Within the context of dynamic FC, it is essential to capture these time varying spatial relationships. Most unsupervised representation learning methods for DBGs tend to focus on clustering DBGs into a finite number of connectivity patterns that recur over time (Allen et al., 2014; Spencer & Goodfellow, 2022). Community detection is another commonly used method but is mainly applied to static brain graphs (Pavlović et al., 2020; Esfahlani et al., 2021). Extensions to DBGs are typically not end-to-end trainable and do not scale to multi-subject datasets (Ting et al., 2020; Martinet et al., 2020a). Recent deep learning-based methods are predominantly GNN-based (Kim et al., 2021; Dahan et al., 2021). Unlike NeuroEvolve, these methods are supervised and focus on learning deterministic node- and/or graph-level

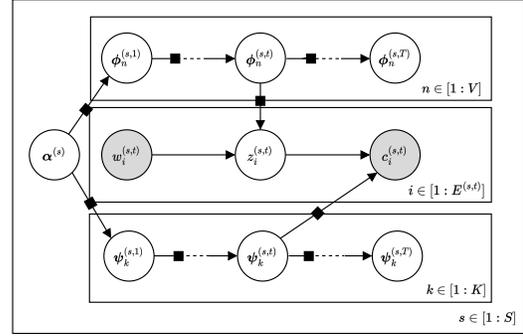


Figure 1. Plate diagram summarizing the generative model of NeuroEvolve. Latent and observed variables are denoted by white- and gray-shaded circles, respectively. Solid black squares denote mappings parameterized by a neural network.

representations.

3. Problem formulation

We consider a dataset of multi-subject DBGs derived from fMRI data $\mathcal{D} \equiv \mathcal{G}^{(1:S, 1:T)} = \{\mathcal{G}^{(s,t)}\}_{s,t=1}^{S,T}$ that share a common set of nodes $\mathcal{V} = \{v_1, \dots, v_V\}$ over $T \in \mathbb{N}$ snapshots for $S \in \mathbb{N}$ subjects. Each $\mathcal{G}^{(s,t)} \in \mathcal{G}^{(1:S, 1:T)}$ denotes a non-attributed, unweighted, and undirected brain graph for the s -th subject at the t -th snapshot. We define a brain graph snapshot as a tuple $\mathcal{G}^{(s,t)} = (\mathcal{V}, \mathcal{E}^{(s,t)})$ where $\mathcal{E}^{(s,t)} \subseteq \mathcal{V} \times \mathcal{V}$ denotes an edge set. The i -th edge for the s -th subject at the t -th snapshot $e_i^{(s,t)} \in \mathcal{E}^{(s,t)}$ is defined $e_i^{(s,t)} = (w_i^{(s,t)}, c_i^{(s,t)})$ where $w_i^{(s,t)}$ is a source node and $c_i^{(s,t)}$ is a target node.

We assume each node corresponds to a brain region making the number of nodes $|\mathcal{V}| = V \in \mathbb{N}$ fixed over subjects and snapshots. We also assume edges correspond to a measure FC allowing the number of edges $|\mathcal{E}^{(s,t)}| = E^{(s,t)} \in \mathbb{N}$ to vary over subjects as well as snapshots. We further assume there exists $K \in \mathbb{N}$ clusters of nodes, or communities, the membership of which dynamically changes between snapshots for each subject. Let $z_i^{(s,t)} \in [1:K]$ denote the latent community assignment of the i -th edge for the s -th subject at the t -th snapshot.

For each subject’s DBG we aim to learn, in an unsupervised fashion, graph $\alpha^{(s)} \in \mathbb{R}^{H_\alpha}$, node $\phi_{1:V}^{(s,t)} = [\phi_n^{(s,t)}] \in \mathbb{R}^{V \times H_\phi}$, and community $\psi_{1:K}^{(s,t)} = [\psi_k^{(s,t)}] \in \mathbb{R}^{K \times H_\psi}$ embeddings of dimensions $H_\alpha, H_\phi, H_\psi \in \mathbb{N}$, respectively, for use in a variety of downstream tasks.

4. Method

NeuroEvolve is a hierarchical deep generative model and inference network that accomplishes end-to-end learning of

¹Code available at <https://github.com/simeon-spasov/dynamic-brain-graph-deep-generative-model>

graph, node, and community embeddings from multi-subject DBG data. NeuroEvolve treats the embeddings and edge community assignments as latent random variables $\Omega^{(s,t)} = \{\alpha^{(s)}, \phi_{1:V}^{(s,t)}, \psi_{1:K}^{(s,t)}, \{z_i^{(s,t)}\}_{i=1}^{E^{(s,t)}}\}$, which along with the observed DBGs, characterize a probabilistic latent variable model with joint density $p_\theta(\mathcal{G}^{1:S, 1:T}, \Omega^{1:S, 1:T})$.

4.1. Generative model

Algorithm 1 NeuroEvolve generative process

```

1: Input: Common node set  $\mathcal{V}$ , source nodes from all edges
    $\{w_i^{(s,t)} : i = 1, \dots, E^{(s,t)}\}_{s,t=1}^{S,T}$ 
2: Require: Number of communities  $K \in \mathbb{N}$ ; embedding dimensions
    $H_\alpha, H_\phi, H_\psi \in \mathbb{N}$ ; number of layers in NNs  $L_\phi, L_\psi,$ 
    $L_z \in \mathbb{N}$ ; temporal smoothness  $\sigma_\psi, \sigma_\phi \in \mathbb{R}_{>0}$ 
3:  $\mathcal{D} \leftarrow \emptyset$ 
4: for  $s \leftarrow 1$  to  $S$  do
5:    $\alpha^{(s)} \sim p(\alpha^{(s)}) = \text{Normal}(\mathbf{0}_{H_\alpha}, \mathbf{I}_{H_\alpha})$ 
6:   for  $t \leftarrow 1$  to  $T$  do
7:     for  $k \leftarrow 1$  to  $K$  do
8:       if  $t = 1$  then
9:          $\psi_k^{(s,0)} = \text{MLP}_{\theta_\psi}(\alpha^{(s)})$ 
10:      end if
11:       $p(\psi_k^{(s,t)} | \psi_k^{(s,t-1)}) = \text{Normal}(\psi_k^{(s,t-1)}, \sigma_\psi \mathbf{I}_{H_\psi})$ 
12:       $\psi_k^{(s,t)} \sim p(\psi_k^{(s,t)} | \psi_k^{(s,t-1)})$ 
13:    end for
14:    for  $n \leftarrow 1$  to  $V$  do
15:      if  $t = 1$  then
16:         $\phi_n^{(s,0)} = \text{MLP}_{\theta_\phi}(\alpha^{(s)})$ 
17:      end if
18:       $p(\phi_n^{(s,t)} | \phi_n^{(s,t-1)}) = \text{Normal}(\phi_n^{(s,t-1)}, \sigma_\phi \mathbf{I}_{H_\phi})$ 
19:       $\phi_n^{(s,t)} \sim p(\phi_n^{(s,t)} | \phi_n^{(s,t-1)})$ 
20:    end for
21:     $\mathcal{E}^{(s,t)} \leftarrow \emptyset$ 
22:    for  $i \leftarrow 1$  to  $E^{(s,t)}$  do
23:       $\text{logit } \tilde{\pi}_i^{(s,t)} = \text{MLP}_{\theta_z}(\phi_{w_i^{(s,t)}}^{(s,t)})$ 
24:       $p(z_i^{(s,t)} | w_i^{(s,t)}) = \text{Categorical}(\tilde{\pi}_i^{(s,t)})$ 
25:       $z_i^{(s,t)} \sim p(z_i^{(s,t)} | w_i^{(s,t)})$ 
26:       $\text{logit } \tilde{\pi}_i^{(s,t)} = \text{MLP}_{\theta_c}(\psi_{z_i^{(s,t)}}^{(s,t)})$ 
27:       $p_{\theta_c}(c_i^{(s,t)} | z_i^{(s,t)}) = \text{Categorical}(\tilde{\pi}_i^{(s,t)})$ 
28:       $c_i^{(s,t)} \sim p_{\theta_c}(c_i^{(s,t)} | z_i^{(s,t)})$ 
29:       $\mathcal{E}^{(s,t)} \leftarrow \mathcal{E}^{(s,t)} \cup \{(w_i^{(s,t)}, c_i^{(s,t)})\}$ 
30:    end for
31:     $\mathcal{G}^{(s,t)} \leftarrow (\mathcal{V}, \mathcal{E}^{(s,t)})$ 
32:     $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathcal{G}^{(s,t)}\}$ 
33:  end for
34: end for

```

Graph embeddings The generative process of NeuroEvolve begins by sampling graph embeddings from a prior $\alpha^{(s)} \sim p_{\theta_\alpha}(\alpha^{(s)})$ implemented as a normal distribution

$$p_{\theta_\alpha}(\alpha^{(s)}) = \text{Normal}(\mathbf{0}_{H_\alpha}, \mathbf{I}_{H_\alpha}) \quad (1)$$

where $\mathbf{0}_{H_\alpha}$ and \mathbf{I}_{H_α} denote a zero matrix and an identity matrix, respectively. Each embedding, represented as a H_α -dimensional vector $\alpha^{(s)} \in \mathbb{R}^{H_\alpha}$, encapsulates subject-specific information that remains constant over snapshots.

Node and community embeddings Next, let $\phi_n^{(s,t)} \in \mathbb{R}^{H_\phi}$ and $\psi_k^{(s,t)} \in \mathbb{R}^{H_\psi}$ denote the n -th node and the k -th community embedding, respectively. To incorporate temporal dynamics, we assume the node and community embeddings are related through Markov chains with prior transition distributions $\phi_n^{(s,t)} \sim p_{\theta_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)})$ and $\psi_k^{(s,t)} \sim p_{\theta_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)})$. We specify each prior to be a normal distribution following

$$p_{\theta_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)}) = \text{Normal}(\phi_n^{(s,t-1)}, \sigma_\phi \mathbf{I}_{H_\phi}) \quad (2)$$

$$p_{\theta_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)}) = \text{Normal}(\psi_k^{(s,t-1)}, \sigma_\psi \mathbf{I}_{H_\psi}). \quad (3)$$

The means of each distribution are initialized via NN transformations of the graph embeddings, i.e. $\phi_n^{(s,0)} = \text{MLP}_{\theta_\phi}(\alpha^{(s)})$, $\psi_k^{(s,0)} = \text{MLP}_{\theta_\psi}(\alpha^{(s)})$, where $\text{MLP}_{\theta_j} : \mathbb{R}^{H_\alpha} \rightarrow \mathbb{R}^{H_j}$ is a L_j -layered multilayer perceptron (MLP) for $j \in \{\phi, \psi\}$. The standard deviations $\sigma_\phi, \sigma_\psi \in \mathbb{R}_{\geq 0}$ are hyperparameters controlling how smoothly each embedding changes between consecutive snapshots.

Edge generation We next describe the edge generative process of a graph snapshot $\mathcal{G}^{(s,t)} \in \mathcal{G}^{(1:S, 1:T)}$. Similar to Sun et al. (2019), for each edge $e_i^{(s,t)} = (w_i^{(s,t)}, c_i^{(s,t)}) \in \mathcal{E}^{(s,t)}$ we first sample a latent community assignment $z_i^{(s,t)} \in [1 : K]$ from a conditional prior $z_i^{(s,t)} \sim p_{\theta_z}(z_i^{(s,t)} | w_i^{(s,t)})$ implemented as a categorical distribution

$$p_{\theta_z}(z_i^{(s,t)} | w_i^{(s,t)}) = \text{Categorical}(\tilde{\pi}_i^{(s,t)}) \quad (4)$$

$$\text{logit } \tilde{\pi}_i^{(s,t)} = \text{MLP}_{\theta_z}(\phi_{w_i^{(s,t)}}^{(s,t)}) \quad (5)$$

where $\text{MLP}_{\theta_z} : \mathbb{R}^{H_\phi} \rightarrow \mathbb{R}^K$ is a L_z -layered MLP that parameterizes community probabilities using node embeddings indexed by $w_i^{(s,t)}$. In other words, each source node $w_i^{(s,t)}$ is represented as a mixture of communities.

A linked target node $c_i^{(s,t)} \in [1 : V]$ is then sampled from the conditional likelihood $c_i^{(s,t)} \sim p_{\theta_c}(c_i^{(s,t)} | z_i^{(s,t)})$ which is also implemented as a categorical distribution

$$p_{\theta_c}(c_i^{(s,t)} | z_i^{(s,t)}) = \text{Categorical}(\hat{\pi}_i^{(s,t)}) \quad (6)$$

$$\text{logit } \hat{\pi}_i^{(s,t)} = \text{MLP}_{\theta_c}(\psi_{z_i^{(s,t)}}^{(s,t)}) \quad (7)$$

where $\text{MLP}_{\theta_c} : \mathbb{R}^{H_\psi} \rightarrow \mathbb{R}^V$ is a L_c -layered MLP that parameterizes node probabilities using community embeddings indexed by $z_i^{(s,t)}$. That is, each community assignment $z_i^{(s,t)}$ is represented as a mixture of nodes. By integrating out the latent community assignment variable

$$p(c_i^{(s,t)} | w_i^{(s,t)}) = \sum_{z_i^{(s,t)} \in [1:K]} p_{\theta_c}(c_i^{(s,t)} | z_i^{(s,t)}) p_{\theta_z}(z_i^{(s,t)} | w_i^{(s,t)}) \quad (8)$$

we define the likelihood of node $c_i^{(s,t)}$ being a linked neighbor of node $w_i^{(s,t)}$, in a given graph snapshot.

Factorized generative model Given this generative process, the joint probability of the observed data and the latent variables can be factorized following

$$\begin{aligned}
 p_\theta(\mathcal{G}^{1:S, 1:T}, \Omega^{1:S, 1:T}) &= \prod_{s=1}^S \left(p_{\theta_\alpha}(\alpha^{(s)}) \right. \\
 &\times \prod_{t=1}^T \left(\prod_{n=1}^V p_{\theta_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)}) \right. \\
 &\times \prod_{k=1}^K p_{\theta_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)}) \quad (9) \\
 &\times \prod_{i=1}^{E^{(s,t)}} p_{\theta_z}(z_i^{(s,t)} | \phi_{w_i}^{(s,t)}) \\
 &\left. \left. \times p_{\theta_c}(c_i^{(s,t)} | \psi_{z_i}^{(s,t)}) \right) \right)
 \end{aligned}$$

where $\theta = \{\theta_\phi, \theta_\psi, \theta_z, \theta_c\}$ is the set of generative model parameters, i.e. NN weights. See Figure 1 for a graphical representation of NeuroEvolve and Algorithm 1 for a summary of the generative process.

4.2. Inference model

Inferring the posterior distribution $p_\theta(\Omega^{(1:S, 1:T)} | \mathcal{G}^{(1:S, 1:T)})$ is intractable so we resort to variational inference (Jordan et al., 1999) to approximate the true posterior with a variational distribution $q_\lambda(\Omega^{(1:S, 1:T)})$. For our training algorithm, we maximize a lower bound on the log marginal likelihood of the DBGs, referred to as the ELBO (evidence lower bound)

$$\begin{aligned}
 \mathcal{L}_{\text{ELBO}}(\theta, \lambda) &= \mathbb{E}_{q_\lambda} \left[\log \frac{p_\theta(\mathcal{G}^{(1:S, 1:T)}, \Omega^{(1:S, 1:T)})}{q_\lambda(\Omega^{(1:S, 1:T)})} \right] \\
 &\leq \log p_\theta(\mathcal{G}^{(1:S, 1:T)}) \quad (10)
 \end{aligned}$$

where $\mathbb{E}_{q_\lambda}[\cdot]$ denotes the expectation with respect to the variational distribution $q_\lambda(\Omega^{(1:S, 1:T)})$. By maximizing the ELBO with respect to the generative and variational parameters θ and λ we train our generative model and perform Bayesian inference, respectively.

Structured variational distribution To ensure a good approximation to the true the posterior, we retain the Markov properties of the node and community embeddings resulting in a structured variational distribution (Hoffman & Blei,

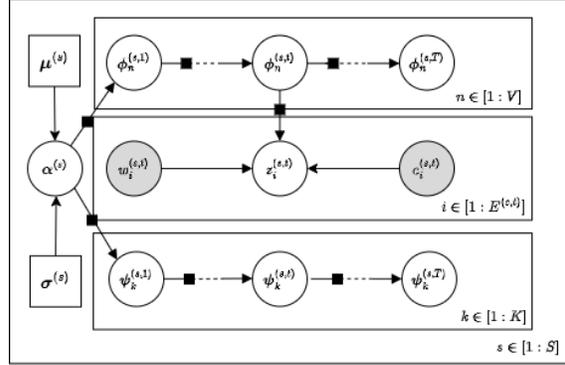


Figure 2. Plate diagram summarizing the inference model of NeuroEvolve. Latent and observed variables are denoted by white- and gray-shaded circles, respectively. Solid black squares denote mappings parameterized by a neural network.

2015; Saul & Jordan, 1995) which factorizes following

$$\begin{aligned}
 q_\lambda(\Omega^{(1:S, 1:T)}) &= \prod_{s=1}^S \left(q_{\lambda_\alpha}(\alpha^{(s)}) \right. \\
 &\times \prod_{t=1}^T \left(\prod_{n=1}^V q_{\lambda_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)}) \right. \\
 &\times \prod_{k=1}^K q_{\lambda_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)}) \quad (11) \\
 &\left. \left. \times \prod_{i=1}^{E^{(s,t)}} q_{\lambda_z}(z_i^{(s,t)} | \phi_{w_i}^{(s,t)}, \phi_{c_i}^{(s,t)}) \right) \right).
 \end{aligned}$$

Moreover, we specify each variational distribution to be from the same family of distribution as it's equivalent in the generative model. For the graph embeddings we have

$$q_{\lambda_\alpha}(\alpha^{(s)}) = \text{Normal}(\mu^{(s)}, \sigma^{(s)} \mathbf{I}_{H_\alpha}). \quad (12)$$

Next, for the node embeddings

$$q_{\lambda_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)}) = \text{Normal}(\hat{\mu}_n^{(s,t)}, \hat{\sigma}_n^{(s,t)} \mathbf{I}_{H_\phi}) \quad (13)$$

$$\{\hat{\mu}_n^{(s,t)}, \log \hat{\sigma}_n^{(s,t)}\} = \text{GRU}_{\lambda_\phi}(\phi_n^{(s,t-1)}) \quad (14)$$

and community embeddings

$$q_{\lambda_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)}) = \text{Normal}(\tilde{\mu}_k^{(s,t)}, \tilde{\sigma}_k^{(s,t)} \mathbf{I}_{H_\psi}) \quad (15)$$

$$\{\tilde{\mu}_k^{(s,t)}, \log \tilde{\sigma}_k^{(s,t)}\} = \text{GRU}_{\lambda_\psi}(\psi_k^{(s,t-1)}) \quad (16)$$

where $\text{GRU}_{\lambda_j} : \mathbb{R}^{H_j} \rightarrow \mathbb{R}^{H_j}$ is a L_j -layered gated recurrent unit (GRU) (Cho et al., 2014) for $j \in \{\phi, \psi\}$. Furthermore, we use MLPs to initialize the GRUs with graph embeddings such that $\phi_n^{(s,0)} = \text{MLP}_{\lambda_\phi}(\alpha^{(s)})$ and $\psi_k^{(s,0)} = \text{MLP}_{\lambda_\psi}(\alpha^{(s)})$ where $\text{MLP}_{\lambda_j} : \mathbb{R}^{N_\alpha} \rightarrow \mathbb{R}^{N_j}$ for $j \in \{\phi, \psi\}$. This allows for subject-specific variation to

be incorporated in the temporal dynamics of the node and community embeddings. Finally, for the latent community assignment of each edge we define

$$q_{\lambda_z}(z_i^{(s,t)} | \phi_{w_i}^{(s,t)}, \phi_{c_i}^{(s,t)}) = \text{Categorical}(\pi_i^{(s,t)}) \quad (17)$$

$$\text{logit } \pi_i^{(s,t)} = \text{MLP}_{\lambda_z}(\phi_{w_i}^{(s,t)} \odot \phi_{c_i}^{(s,t)}) \quad (18)$$

where $\text{MLP}_{\lambda_z} : \mathbb{R}^{H_\phi} \rightarrow \mathbb{R}^K$ is L_z -layered MLP. In contrast to the generative model, the variational distribution of the community assignment now includes information from neighboring nodes via $c_i^{(s,t)}$. See Figure 2 for a graphical representation of the inference model.

Training objective Substituting the variational distribution from (11) and the joint distribution from (9) into the ELBO (10) gives the full training objective, which for the s -th subject is defined as

$$\begin{aligned} \mathcal{L}_{\text{ELBO}}^{(s)}(\theta, \lambda) = & \sum_{t=1}^T \sum_{i=1}^{E^{(s,t)}} \left(\mathbb{E}_{q_{\lambda_z} q_{\lambda_\psi}} \left[\log p_{\theta_c}(c_i^{(s,t)} | w_i^{(s,t)}, \psi_{z_i}^{(s,t)}) \right] \right. \\ & \left. - \mathbb{E}_{q_{\lambda_\phi}} \left[\text{D}_{\text{KL}}[q_{\lambda_z}(z_i^{(s,t)} | \phi_{w_i}^{(s,t)}, \phi_{c_i}^{(s,t)}) || \right. \right. \\ & \left. \left. p_{\theta_z}(z_i^{(s,t)} | \phi_{w_i}^{(s,t)}) \right] \right) \\ & - \text{D}_{\text{KL}}[q_{\lambda_\alpha}(\alpha^{(s)}) || p_{\theta_\alpha}(\alpha^{(s)})] \sum_{t=1}^T \left(\right. \\ & - \sum_{n=1}^V \mathbb{E}_{q_{\lambda_\phi}} \left[\text{D}_{\text{KL}}[q_{\lambda_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)}) || \right. \\ & \left. \left. p_{\theta_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)}) \right] \right) \\ & - \sum_{k=1}^K \mathbb{E}_{q_{\lambda_\psi}} \left[\text{D}_{\text{KL}}[q_{\lambda_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)}) || \right. \\ & \left. \left. p_{\theta_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)}) \right] \right) \end{aligned} \quad (19)$$

where $\text{D}_{\text{KL}}[\cdot || \cdot]$ denotes the Kullback-Leibler (KL) divergence. By maximizing the ELBO, the parameters (θ, λ) of the generative model and inference model can be jointly learnt. See Figure 2 for a graphical representation of the inference model of NeuroEvolve.

Algorithm 2 NeuroEvolve inference model

```

1: Input: Common node set  $\mathcal{V}$ , source nodes of all edges
    $\{w_i^{(s,t)} : i = 1, \dots, E^{(s,t)}\}_{s,t=1}^{S,T}$ 
2: Require: Number of communities  $K \in \mathbb{N}$ ; embedding dimensions
    $H_\alpha, H_\phi, H_\psi \in \mathbb{N}$ ; number of layers in NNs  $L_\phi, L_\psi$ 
    $L_z \in \mathbb{N}$ 
3:  $\mathcal{L} \leftarrow 0$ 
4:  $\{\mu^{(s)}, \log \sigma^{(s)}\}_{s=1}^S \leftarrow \text{Normal}(\mathbf{0}_{H_\alpha}, \mathbf{I}_{H_\alpha})$ 
5: repeat
6:   for  $s \leftarrow \text{RandomShuffle}[1 : S]$  do
7:      $q_{\lambda_\alpha}(\alpha^{(s)}) = \text{Normal}(\mu^{(s)}, \sigma^{(s)} \mathbf{I}_{H_\alpha})$ 
8:      $\alpha^{(s)} \sim q_{\lambda_\alpha}(\alpha^{(s)})$ 
9:     for  $t \leftarrow 1$  to  $T$  do
10:      for  $k \leftarrow 1$  to  $K$  do
11:        if  $t = 1$  then
12:           $\psi_k^{(s,0)} = \text{MLP}_{\lambda_\psi}(\alpha^{(s)})$ 
13:        end if
14:         $\{\tilde{\mu}_k^{(s,t)}, \log \tilde{\sigma}_k^{(s,t)}\} = \text{GRU}_{\lambda_\psi}(\psi_k^{(s,t-1)})$ 
15:         $q_{\lambda_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)}) =$ 
            $\text{Normal}(\tilde{\mu}_k^{(s,t)}, \tilde{\sigma}_k^{(s,t)} \mathbf{I}_{H_\psi})$ 
16:         $\psi_k^{(s,t)} \sim q_{\lambda_\psi}(\psi_k^{(s,t)} | \psi_k^{(s,t-1)})$ 
17:      end for
18:      for  $n \leftarrow 1$  to  $V$  do
19:        if  $t = 1$  then
20:           $\phi_n^{(s,0)} = \text{MLP}_{\lambda_\phi}(\alpha^{(s)})$ 
21:        end if
22:         $\{\tilde{\mu}_n^{(s,t)}, \log \tilde{\sigma}_n^{(s,t)}\} = \text{GRU}_{\lambda_\phi}(\phi_n^{(s,t-1)})$ 
23:         $q_{\lambda_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)}) =$ 
            $\text{Normal}(\tilde{\mu}_n^{(s,t)}, \tilde{\sigma}_n^{(s,t)} \mathbf{I}_{H_\phi})$ 
24:         $\phi_n^{(s,t)} \sim q_{\lambda_\phi}(\phi_n^{(s,t)} | \phi_n^{(s,t-1)})$ 
25:      end for
26:      for  $i \leftarrow 1$  to  $E^{(s,t)}$  do
27:         $\text{logit } \hat{\pi}_i^{(s,t)} = \text{MLP}_{\lambda_z}(\phi_{w_i}^{(s,t)} \odot \phi_{c_i}^{(s,t)})$ 
28:         $q_{\lambda_z}(z_i^{(s,t)} | \phi_{w_i}^{(s,t)}, \phi_{c_i}^{(s,t)}) = \text{Categorical}(\hat{\pi}_i^{(s,t)})$ 
29:         $z_i^{(s,t)} \sim q_{\lambda_z}(z_i^{(s,t)} | \phi_{w_i}^{(s,t)}, \phi_{c_i}^{(s,t)})$ 
30:         $\text{logit } \tilde{\pi}_i^{(s,t)} = \text{MLP}_{\theta_c}(\psi_{z_i}^{(s,t)})$ 
31:         $p_{\theta_c}(c_i^{(s,t)} | z_i^{(s,t)}) = \text{Categorical}(\tilde{\pi}_i^{(s,t)})$ 
32:         $c_i^{(s,t)} \sim p_{\theta_c}(c_i^{(s,t)} | z_i^{(s,t)})$ 
33:      end for
34:    end for
35:    Compute gradients of  $\mathcal{L}_{\text{ELBO}}^{(s)}(\theta, \lambda)$  w.r.t.  $(\theta, \lambda)$ 
36:    Perform gradient-based updates for  $(\theta, \lambda)$ 
37:     $\mathcal{L} \leftarrow \mathcal{L} + \frac{1}{S} \mathcal{L}_{\text{ELBO}}^{(s)}(\theta, \lambda)$ 
38:  end for
39: until  $\mathcal{L}$  converges
    
```

Inference and learning In order to use efficient stochastic gradient-based optimization techniques (Robbins & Monro, 1951) for learning the parameters, the gradient of the ELBO (19) must be estimated with respect to (θ, λ) . The main challenge is obtaining gradients of the variables under expectation, i.e. $\mathbb{E}_{q_*}[\cdot]$, since they are stochastic. To allow gradients to pass through these sampling steps, we use the reparameterization trick (Kingma & Welling, 2013; Rezende et al., 2014) for the normal distributions and the Gumbel-softmax trick (Jang et al., 2016; Maddison et al., 2016)

Table 1. Results for graph reconstruction (top) and dynamic link prediction (bottom). First and second-best results shown in **bold** and underlined. Results with a statistically significant difference from NeuroEvolve are marked *.

Model	HCP		UKB	
	NLL (\downarrow)	MSE (\downarrow)	NLL (\downarrow)	MSE (\downarrow)
VGAE	5.857 \pm 0.017 *	0.051 \pm 0.002 *	5.851 \pm 0.027 *	0.061 \pm 0.002 *
OSBM	5.808 \pm 0.026 *	0.051 \pm 0.003 *	5.726 \pm 0.039 *	0.052 \pm 0.003 *
VGRAPH	<u>5.569 \pm 0.046 *</u>	0.022 \pm 0.004 *	5.716 \pm 0.037 *	0.020 \pm 0.003 *
VGRNN	<u>5.674 \pm 0.034 *</u>	<u>0.011 \pm 0.003 *</u>	<u>5.649 \pm 0.035 *</u>	<u>0.014 \pm 0.002 *</u>
ELSM	5.924 \pm 0.040 *	0.081 \pm 0.002 *	5.809 \pm 0.024 *	0.115 \pm 0.003 *
NeuroEvolve	4.587 \pm 0.045	0.001 \pm 0.002	4.586 \pm 0.084	0.004 \pm 0.003
	AUROC (\uparrow)	AP (\uparrow)	AUROC (\uparrow)	AP (\uparrow)
VGAE	0.661 \pm 0.010 *	0.674 \pm 0.008 *	0.688 \pm 0.010 *	0.607 \pm 0.009 *
OSBM	0.655 \pm 0.027 *	0.675 \pm 0.024 *	0.678 \pm 0.032 *	0.682 \pm 0.033 *
VGRAPH	<u>0.689 \pm 0.004 *</u>	0.682 \pm 0.002 *	0.664 \pm 0.002 *	0.621 \pm 0.001 *
VGRNN	<u>0.689 \pm 0.007 *</u>	<u>0.698 \pm 0.006 *</u>	<u>0.698 \pm 0.009 *</u>	<u>0.696 \pm 0.007 *</u>
ELSM	0.669 \pm 0.004 *	0.662 \pm 0.002 *	0.661 \pm 0.001 *	0.662 \pm 0.002 *
NeuroEvolve	0.768 \pm 0.026	0.732 \pm 0.032	0.786 \pm 0.040	0.762 \pm 0.038

for the categorical distributions. This allows for the gradient of (19) w.r.t. θ and λ to be easily computed via back-propagation (Rumelhart et al., 1986) making NeuroEvolve end-to-end trainable. In addition, we analytically calculate the KL terms for both normal and categorical distributions, which leads to lower variance gradient estimates and faster training as compared to noisy Monte Carlo estimates.

Parameter sharing We use the same NNs from the generative model to parameterize the variational distributions for the node and community embeddings as well as the edge community assignments. This not only spares additional trainable parameters for the variational distribution but also further links the variational parameters of $q_\lambda(\cdot)$ to generative parameters of $p_\theta(\cdot)$ resulting in more robust learning (Farnoosh & Ostadabbas, 2021). The set of parameters for the inference network is therefore $\lambda = \{\lambda_\alpha = \{\mu^{(s)}, \sigma^{(s)}\}_{s=1}^S, \lambda_\phi = \theta_\phi, \lambda_\psi = \theta_\psi, \lambda_z = \theta_z\}$.

5. Experiments

We evaluate NeuroEvolve against a range of unsupervised probabilistic graph representation learning baseline models on the tasks of graph reconstruction, dynamic link prediction, and graph classification. A good graph representation learning method should be able to preserve most of the original graph structure in its embeddings. Since NeuroEvolve learns a hierarchy of embeddings, we use each task to quantitatively assess its ability at preserving different levels of graph structure compared to baselines.

Datasets We construct two multi-subject DBG datasets using publicly available fMRI data from the Human Connectome Project (HCP) (Van Essen et al., 2013) and UK Biobank (UKB) (Sudlow et al., 2015). Both data sources represent well-characterized population cohorts that have undergone standardized neuroimaging and clinical assessments to ensure high quality. We randomly sample $S = 300$ subjects from each dataset ensuring an even split in biological sex. To create DBGs, we parcellate each image into $V = 360$ BOLD signals using the Glasser atlas (Glasser et al., 2016), apply sliding-window Pearson correlation (Calhoun et al., 2014) with a non-overlapping window of size and stride 30, and threshold the top 5% values of each correlation matrix as connected following Kim et al. (2021). This results in $T = 16$ graph snapshots for each subject. Biological sex is taken as graph-level labels. We refer to Appendix C for further details on each dataset.

Baselines We compare NeuroEvolve against a range of different unsupervised probabilistic graph representation learning baseline models. For static baselines, we include variational graph autoencoder (VGAE) (Kipf & Welling, 2016b), a deep generative version of the overlapping stochastic block model (OSBM) (Mehta et al., 2019), and vGraph (VGRAPH) (Sun et al., 2019). For dynamic baselines we include variational graph recurrent neural network (VGRNN) (Hajiramezanali et al., 2019) and evolving latent space model (ELSM) (Gupta et al., 2019). Finally, for graph classification we include a support vector machine which takes as inputs static FC matrices (FCM) (Abraham et al., 2017). Further details about baseline models can be found in Appendix D.

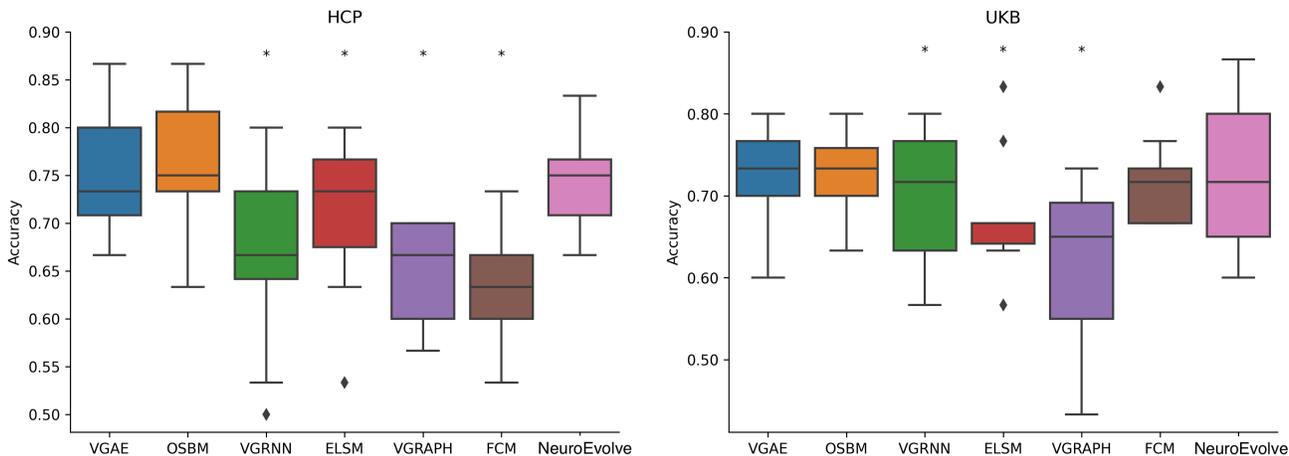


Figure 3. Graph classification results (5 runs). Results with a statistically significant difference from NeuroEvolve are marked *.

Implementation We split both datasets into 80/10/10% training/validation/test data along the snapshot dimension. All models are trained for 1,000 epochs using the Adam optimizer (Kingma & Ba, 2014) with decoupled weight decay (Loshchilov & Hutter, 2017). For static graph baselines VGAE, OSBM, VGRAPH we train on each snapshot independently and use node/community embeddings at the last training snapshot to make predictions. All models are trained 5 times using different random seeds and the model with the lowest validation negative log-likelihood (NLL) is saved. See Appendix E for further implementation details.

Evaluation metrics For graph reconstruction, we calculate the probability of observing edges over test snapshots using NLL and also compare the mean-squared error (MSE) between actual and reconstructed node degree over all snapshots. For dynamic link prediction, we sample an equal number of positive and negative edges and measure performance using area under the receiver operator curve (AUROC) and average precision (AP). Finally, for graph classification, we predict the biological sex for each subject’s DBG and evaluate on accuracy. To do this, we average per-subject node-level embeddings for the baseline models and use the graph-level embeddings for NeuroEvolve before training a support vector machine (Murphy, 2012) using 10-fold cross-validation. Finally, for comparing results we use the almost stochastic order (ASO) test (Del Barrio et al., 2018; Dror et al., 2019) with significance level 0.05 and correct for multiple comparisons (Bonferroni, 1936).

6. Results

Graph reconstruction and dynamic link prediction Table 1 summarizes the test results averaged over 5 runs. From the results, it is clear that on both tasks NeuroEvolve outper-

forms all baselines by statistically significant margins. In particular, for graph reconstruction NeuroEvolve achieves an 18% and 30% relative improvement in NLL on HCP and UKB compared to the second-best baselines, respectively. For dynamic link prediction, the relative improvement of NeuroEvolve is $> 11\%$ in AUROC and $> 5\%$ in AP compared to second-best baselines depending on dataset.

Graph classification For graph classification, NeuroEvolve achieves $\sim 75\%$ accuracy for HCP and $\sim 73\%$ for UKB (see Fig. 3). We outperform 4 baselines and show indiscernible performance to VGAE and OSBM. To show the interpretative power of NeuroEvolve, we re-run the graph classification experiment for HCP with the embeddings of each community separately. We find a community which comprises brain regions in the Cingulo-opercular (CON) and the Somatomotor (SMN) networks, which achieves 68% accuracy. This finding is in agreement with studies that show SMN is predictive of gender (Zhang et al., 2018). With the exception of VGRAPH, which NeuroEvolve outperforms, such an interpretability analysis cannot be done in a computationally feasible way by any of the other baselines.

7. Interpretability analysis

Evidence from fMRI studies suggests complex community structures exist within DBGs (Ting et al., 2020; Martinet et al., 2020b). These communities often correspond to groups of anatomically neighboring and/or functionally related brain regions that are engaged in specialized information processing.

Community overlap In order to interpret the community embeddings learnt by NeuroEvolve, for the k -th community embedding we create a node score vector $\vec{\psi}_k \in [0, 1]^V$ by

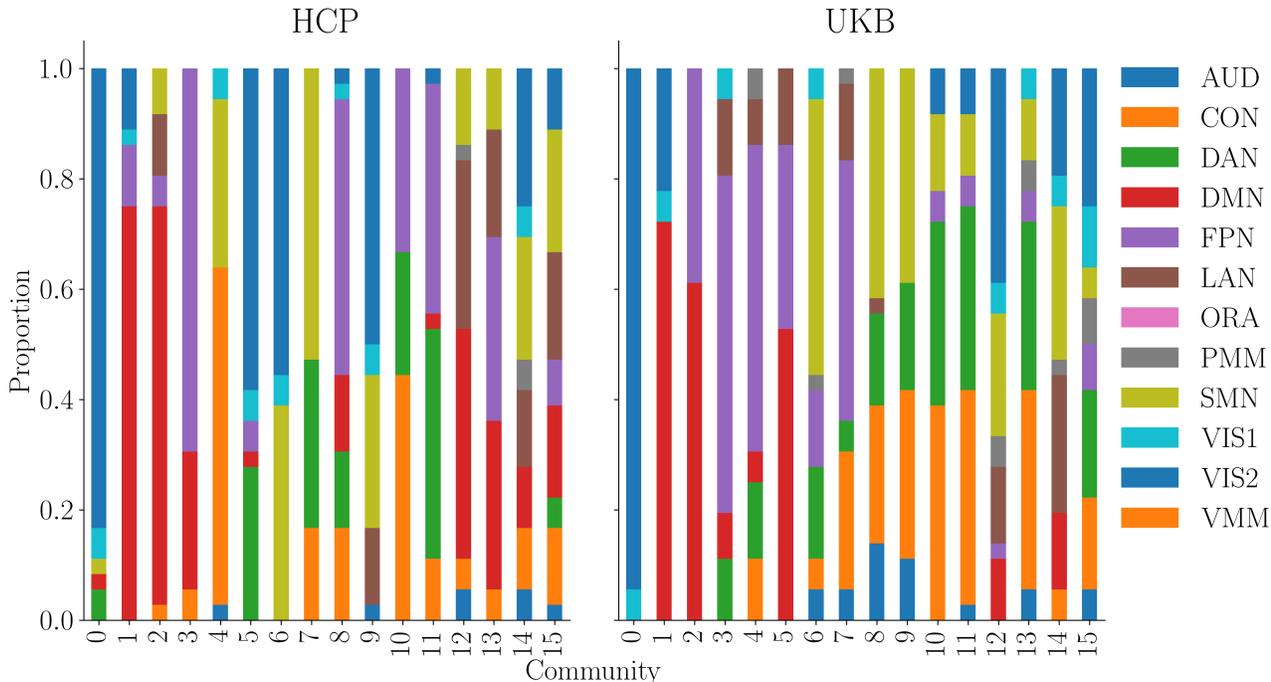


Figure 4. Overlap between node communities learned by NeuroEvolve and FCNs from Ji et al. (2019).

averaging sampled community embeddings over subjects and snapshots following

$$\bar{\psi}_k = \frac{1}{ST} \sum_{s=1}^S \sum_{t=1}^T \text{Softmax}(\text{MLP}_{\lambda_c}(\psi_k^{(s,t)})). \quad (20)$$

We keep the top 10% highest scoring nodes in each score vector and compare their node composition, in terms of proportion of overlap, with known communities from neuroscience literature.

Figure 4 shows the proportion of overlap between nodes in each community and the nodes from FCNs described in Ji et al. (2019) (see Appendix B). It is clear that NeuroEvolve finds communities consisting of nodes that significantly overlap with existing FCNs. In particular, across HCP and UKB nodes in community labelled “0” almost fully correspond to the visual network (VIS1 and VIS2). This is in keeping with the image acquisition protocol of both datasets: subjects were required to keep their eyes open and fixed on a cross-hair. Remarkably, communities “1” and “2”, the second and third most homogeneous communities across both datasets, corresponds largely to the default mode network (DMN), which is well known to dominate resting-state activity (Yeshurun et al., 2021). The inspection of additional communities, along with their evolution over time, has the potential to unveil the relationship between dynamic brain connectivity changes and brain disorders (Heitmann & Breakspear, 2017).

8. Conclusion

We propose NeuroEvolve, a hierarchical deep generative model designed for unsupervised representation learning of DBGs derived from fMRI data. Specifically, NeuroEvolve jointly learns distributions over graph-, community-, and node-level embeddings that evolve over time. Using these embeddings, NeuroEvolve is able to significantly outperform state-of-the-art baselines on the tasks of graph reconstruction and dynamic link prediction. Moreover, an analysis of the learnt dynamic community-node distributions shows significant overlap with existing FCNs from neuroscience literature further validating our method.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and valuable feedback, which greatly contributed to improving the quality of this work. This research was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1. The data was provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

References

- Abraham, A., Milham, M. P., Di Martino, A., Craddock, R. C., Samaras, D., Thirion, B., and Varoquaux, G. Deriving reproducible biomarkers from multi-site resting-state data: An autism-based example. *NeuroImage*, 147:736–745, 2017.
- Adhikari, B., Zhang, Y., Ramakrishnan, N., and Prakash, B. A. Distributed representations of subgraphs. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 111–117. IEEE, 2017.
- Alfaro-Almagro, F., Jenkinson, M., Bangerter, N. K., Andersson, J. L., Griffanti, L., Douaud, G., Sotiropoulos, S. N., Jbabdi, S., Hernandez-Fernandez, M., Vallee, E., et al. Image processing and quality control for the first 10,000 brain imaging datasets from uk biobank. *Neuroimage*, 166:400–424, 2018.
- Allen, E. A., Damaraju, E., Plis, S. M., Erhardt, E. B., Eichele, T., and Calhoun, V. D. Tracking whole-brain connectivity dynamics in the resting state. *Cerebral cortex*, 24(3):663–676, 2014.
- Alonso Martínez, S., Deco, G., Ter Horst, G. J., and Cabral, J. The dynamics of functional brain networks associated with depressive symptoms in a nonclinical sample. *Frontiers in neural circuits*, 14:570583, 2020.
- Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- Bonferroni, C. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.
- Calhoun, V. D., Miller, R., Pearlson, G., and Adalı, T. The chronnectome: time-varying connectivity networks as the next frontier in fmri data discovery. *Neuron*, 84(2):262–274, 2014.
- Cavallari, S., Zheng, V. W., Cai, H., Chang, K. C.-C., and Cambria, E. Learning community embedding with community detection and node embedding on graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 377–386, 2017.
- Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.
- Dahan, S., Williams, L. Z., Rueckert, D., and Robinson, E. C. Improving phenotype prediction using long-range spatio-temporal dynamics of functional connectivity. In *International Workshop on Machine Learning in Clinical Neuroimaging*, pp. 145–154. Springer, 2021.
- Dautricourt, S., Gonneaud, J., Landeau, B., Calhoun, V. D., de Flores, R., Poinsel, G., Bougacha, S., Ourry, V., Touron, E., Kuhn, E., et al. Dynamic functional connectivity patterns associated with dementia risk. *Alzheimer’s Research & Therapy*, 14(1):1–13, 2022.
- Del Barrio, E., Cuesta-Albertos, J. A., and Matrán, C. An optimal transportation approach for assessing almost stochastic order. In *The Mathematics of the Uncertain*, pp. 33–44. Springer, 2018.
- Dror, R., Shlomov, S., and Reichart, R. Deep dominance - how to properly compare deep neural models. In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 2773–2785. Association for Computational Linguistics, 2019.
- Erdos, P., Rényi, A., et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5(1):17–60, 1960.
- Esfahlani, F. Z., Jo, Y., Puxeddu, M. G., Merritt, H., Tanner, J. C., Greenwell, S., Patel, R., Faskowitz, J., and Betzel, R. F. Modularity maximization as a flexible and generic framework for brain network exploratory analysis. *Neuroimage*, 244:118607, 2021.
- Farnoosh, A. and Ostadabbas, S. Deep markov factor analysis: Towards concurrent temporal and spatial analysis of fmri data. *Advances in Neural Information Processing Systems*, 34:17876–17888, 2021.
- Gao, Z., Fu, G., Ouyang, C., Tsutsui, S., Liu, X., Yang, J., Gessner, C., Foote, B., Wild, D., Ding, Y., et al. edge2vec: Representation learning using edge semantics for biomedical knowledge discovery. *BMC bioinformatics*, 20(1):1–15, 2019.
- Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., et al. The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124, 2013.
- Glasser, M. F., Coalson, T. S., Robinson, E. C., Hacker, C. D., Harwell, J., Yacoub, E., Ugurbil, K., Andersson, J., Beckmann, C. F., Jenkinson, M., et al. A multi-modal parcellation of human cerebral cortex. *Nature*, 536(7615):171–178, 2016.
- Goyal, P. and Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

- Goyal, P., Kamra, N., He, X., and Liu, Y. Dyngem: Deep embedding method for dynamic graphs. *arXiv preprint arXiv:1805.11273*, 2018.
- Goyal, P., Chhetri, S. R., and Canedo, A. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*, 187: 104816, 2020.
- Gracious, T., Gupta, S., Kanthali, A., Castro, R. M., and Dukkipati, A. Neural latent space model for dynamic networks and temporal knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4054–4062, 2021.
- Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.
- Guan, S., Jiang, R., Bian, H., Yuan, J., Xu, P., Meng, C., and Biswal, B. The profiles of non-stationarity and non-linearity in the time series of resting-state brain networks. *Frontiers in Neuroscience*, 14:493, 2020.
- Gupta, S., Sharma, G., and Dukkipati, A. A generative model for dynamic networks with applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 7842–7849, 2019.
- Hajiramezanali, E., Hasanzadeh, A., Narayanan, K., Duffield, N., Zhou, M., and Qian, X. Variational graph recurrent neural networks. *Advances in neural information processing systems*, 32, 2019.
- Hamilton, W. L. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2649-2. URL <https://doi.org/10.1038/s41586-020-2649-2>.
- Heitmann, S. and Breakspear, M. Putting the ‘dynamic’ back into dynamic functional connectivity, 181313, 2017.
- Hess, A., Hinz, R., Keliris, G. A., and Boehm-Sturm, P. On the usage of brain atlases in neuroimaging research. *Molecular Imaging and Biology*, 20:742–749, 2018.
- Hirsch, F. and Wohlschlaeger, A. Graph analysis of non-linear fmri connectivity dynamics reveals distinct brain network configurations for integrative and segregated information processing. *Nonlinear Dynamics*, 108(4):4287–4299, 2022.
- Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- Hoffman, M. D. and Blei, D. M. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, pp. 361–369, 2015.
- Huettel, S. A., Song, A. W., McCarthy, G., et al. *Functional magnetic resonance imaging*, volume 1. Sinauer Associates Sunderland, 2004.
- Hutchison, R. M., Womelsdorf, T., Allen, E. A., Bandettini, P. A., Calhoun, V. D., Corbetta, M., Della Penna, S., Duyn, J. H., Glover, G. H., Gonzalez-Castillo, J., et al. Dynamic functional connectivity: promise, issues, and interpretations. *Neuroimage*, 80:360–378, 2013.
- Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- Ji, J. L., Spronk, M., Kulkarni, K., Repovš, G., Anticevic, A., and Cole, M. W. Mapping the human brain’s cortical-subcortical functional network organization. *Neuroimage*, 185:35–57, 2019.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
- Khan, R. A., Anwaar, M. U., Kaddah, O., Han, Z., and Kleinsteuber, M. Unsupervised learning of joint embeddings for node representation and community detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 19–35. Springer, 2021.
- Kim, B.-H., Ye, J. C., and Kim, J.-J. Learning dynamic graph representation of brain connectome with spatio-temporal attention. *Advances in Neural Information Processing Systems*, 34:4314–4327, 2021.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016a.

- Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016b.
- Lawrence, R. M., Bridgeford, E. W., Myers, P. E., Arvapalli, G. C., Ramachandran, S. C., Pisner, D. A., Frank, P. F., Lemmer, A. D., Nikolaidis, A., and Vogelstein, J. T. Standardizing human brain parcellations. *Scientific data*, 8(1): 78, 2021.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.
- Mahdavi, S., Khoshraftar, S., and An, A. dynnode2vec: Scalable dynamic network embedding. In *2018 IEEE international conference on big data (Big Data)*, pp. 3762–3765. IEEE, 2018.
- Martinet, L.-E., Kramer, M., Viles, W., Perkins, L., Spencer, E., Chu, C., Cash, S., and Kolaczyk, E. Robust dynamic community detection with applications to human brain functional networks. *Nature communications*, 11(1):1–13, 2020a.
- Martinet, L.-E., Kramer, M. A., Viles, W., Perkins, L. N., Spencer, E., Chu, C. J., Cash, S. S., and Kolaczyk, E. D. Robust dynamic community detection with applications to human brain functional networks. *Nature Communications*, 11(1):2785, 2020b.
- Mehta, N., Duke, L. C., and Rai, P. Stochastic blockmodels meet graph neural networks. In *International Conference on Machine Learning*, pp. 4466–4474. PMLR, 2019.
- Miller, K., Jordan, M., and Griffiths, T. Nonparametric latent feature models for link prediction. *Advances in neural information processing systems*, 22, 2009.
- Murphy, K. P. *Machine learning: a probabilistic perspective*, chapter 14.5.2, pp. 498–504. MIT press, 2012.
- Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., and Jaiswal, S. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.
- Nowicki, K. and Snijders, T. A. B. Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association*, 96(455):1077–1087, 2001.
- Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., and Leiserson, C. Evolvegn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5363–5370, 2020.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.
- Pavlović, D. M., Guillaume, B. R., Afyouni, S., and Nichols, T. E. Multi-subject stochastic blockmodels with mixed effects for adaptive analysis of individual differences in human brain network cluster structure. *Statistica Neerlandica*, 74(3):363–396, 2020.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- Python Core Team. *Python: A dynamic, open source programming language*. Python Software Foundation, 2019. URL <https://www.python.org/>. Python version 3.7.
- Raz, G., Touroutoglou, A., Wilson-Mendenhall, C., Gilam, G., Lin, T., Gonen, T., Jacob, Y., Atzil, S., Admon, R., Bleich-Cohen, M., et al. Functional connectivity dynamics during film viewing reveal common networks for different emotional experiences. *Cognitive, Affective, & Behavioral Neuroscience*, 16(4):709–723, 2016.
- Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.
- Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- Sankar, A., Wu, Y., Gou, L., Zhang, W., and Yang, H. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, pp. 519–527, 2020.
- Sarkar, A., Mehta, N., and Rai, P. Graph representation learning via ladder gamma variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5604–5611, 2020.
- Saul, L. and Jordan, M. Exploiting tractable substructures in intractable networks. *Advances in neural information processing systems*, 8, 1995.

- Spencer, A. P. and Goodfellow, M. Using deep clustering to improve fmri dynamic functional connectivity analysis. *NeuroImage*, 257:119288, 2022.
- Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *Plos med*, 12(3): e1001779, 2015.
- Sun, F.-Y., Qu, M., Hoffmann, J., Huang, C.-W., and Tang, J. vgraph: A generative model for joint community detection and node representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- Thorndike, R. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.
- Ting, C.-M., Samdin, S. B., Tang, M., and Ombao, H. Detecting dynamic community structure in functional brain networks across individuals: a multilayer approach. *IEEE Transactions on Medical Imaging*, 40(2):468–480, 2020.
- Ulmer, D., Hardmeier, C., and Frellsen, J. deep-significance-easy and meaningful statistical significance testing in the age of neural networks. *arXiv preprint arXiv:2204.06815*, 2022.
- Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., Consortium, W.-M. H., et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.
- Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., and Yang, S. Community preserving network embedding. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.
- Yeshurun, Y., Nguyen, M., and Hasson, U. The default mode network: where the idiosyncratic self meets the shared social world. *Nature Reviews Neuroscience*, 22(3):181–192, 2021.
- Yu, Q., Erhardt, E. B., Sui, J., Du, Y., He, H., Hjelm, D., Cetin, M. S., Rachakonda, S., Miller, R. L., Pearlson, G., and Calhoun, V. D. Assessing dynamic brain graphs of time-varying connectivity in fmri data: Application to healthy controls and patients with schizophrenia. *NeuroImage*, 107:345–355, 2015. ISSN 1053-8119. doi: <https://doi.org/10.1016/j.neuroimage.2014.12.020>. URL <https://www.sciencedirect.com/science/article/pii/S105381191401012X>.
- Zhang, C., Dougherty, C. C., Baum, S. A., White, T., and Michael, A. M. Functional connectivity predicts gender: Evidence for gender differences in resting brain connectivity. *Human brain mapping*, 39(4):1765–1776, 2018.
- Zhang, W., Zhang, L., Pfoser, D., and Zhao, L. Disentangled dynamic graph deep generation. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 738–746. SIAM, 2021.
- Zhou, L., Yang, Y., Ren, X., Wu, F., and Zhuang, Y. Dynamic network embedding by modeling triadic closure process. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

A. Notation

Table 2. Summary of notation.

Notation	Description
$S \in \mathbb{N}$	Number of subjects.
$T \in \mathbb{N}$	Number of snapshots.
$\mathcal{G}^{(1:S, 1:T)} = \{\mathcal{G}^{(s,t)}\}_{s,t=1}^{S,T}$	Multi-subject dynamic brain graph (DBG) dataset derived from functional magnetic resonance imaging (fMRI).
$\mathcal{G}^{(s,t)} = (\mathcal{V}, \mathcal{E}^{(s,t)})$	DBG of the s -th subject at the t -th snapshot.
$\mathcal{V} = \{v_1, \dots, v_V\}$	Set of common nodes.
$V \in \mathbb{N}$	Number of nodes.
$\mathcal{E}^{(s,t)} \subseteq \mathcal{V} \times \mathcal{V}$	Edge set.
$(w_i^{(s,t)}, c_i^{(s,t)}) \in \mathcal{E}^{(s,t)}$	Source node and target node of the i -th edge.
$E^{(s,t)} \in \mathbb{N}$	Number of edges.
$K \in \mathbb{N}$	Number of communities.
$\alpha^{(s)} \in \mathbb{R}^{H_\alpha}$	Subject embedding of dimensionality H_α .
$\phi_n^{(s,t)} \in \mathbb{R}^{H_\phi}$	Node embedding of dimensionality H_ϕ .
$\psi_k^{(s,t)} \in \mathbb{R}^{H_\psi}$	Community embedding of dimensionality H_ψ .
$z_i^{(s,t)} \in [1 : K]$	Community assignment for the i -th edge.
$\Omega^{(s,t)} = \{\alpha^{(s)}, \phi^{(s,t)}, \psi^{(s,t)}, \{z_i^{(s,t)}\}_{i=1}^{E^{(s,t)}}\}$	Set of latent variables.
$p_\theta(\mathcal{G}^{(1:S, 1:T)}, \Omega^{(1:S, 1:T)})$	Joint distribution of observed DBG and unobserved latent variables, i.e. generative model with parameters θ .
$q_\lambda(\Omega^{(1:S, 1:T)} \mathcal{G}^{(1:S, 1:T)})$	Approximate posterior distribution, i.e. inference model with parameters λ .
$\sigma_j \in \mathbb{R}_{\geq 0}$	Temporal smoothness hyperparameter for $j \in \{\phi, \psi\}$.
$\text{MLP}_{\theta_*}(\cdot)$	Multilayered perception (MLP) with L_{θ_*} layers and parameters θ_* .
$\text{GRU}_{\theta_*}(\cdot)$	Gated recurrent unit (GRU) with L_{θ_*} layers and parameters θ_* .

B. Functional connectivity networks

Abbreviation	Functional connectivity network
AUD	Auditory network
CON	Cingulo-opercular network
DAN	Dorsal-attention network
DMN	Default mode network
FPN	Frontoparietal network
LAN	Language network
ORA	Orbito-affective network
PMM	Posterior-multimodal network
SMN	Somatomotor network
VIS1	Visual network 1
VIS2	Visual network 2
VMM	Ventral-multimodal network

Table 3. Functional connectivity networks (FCNs) from Ji et al. (2019).

C. Datasets

To create multi-subject DBG datasets, we use real fMRI scans of the brain from the UK Biobank (Sudlow et al., 2015) and Human Connectome Project (Van Essen et al., 2013). Both data sources represent well-characterized population cohorts that have undergone standardized neuroimaging and clinical assessments to ensure high quality.

UK Biobank² (UKB) The UKB dataset consists of $S = 300$ resting-rate fMRI scans (i.e. 3D image of the brain taken over consecutive timepoints) randomly sampled from the v1.3 January 2017 release ensuring an equal male/female split (i.e. sex balanced) with an age range of 44 – 57 years. The total number of images for each scan is 490 timepoints (6 minutes duration with a repetition time of 0.74s). The dataset is minimally preprocessed following the pipeline described in Alfaro-Almagro et al. (2018).

Human Connectome Project³ (HCP) The HCP dataset similarly consists of $S = 300$ sex balanced resting-state fMRI scans randomly sampled from the S1200 release with an age range of 22 – 35 years. Only images from the first scanning-session using left-right phase encoding are used. The total number of images for each scan is 1,200 timepoints (15 minutes duration with a repetition time of 0.72s). The dataset is minimally preprocessed following the pipeline described in Glasser et al. (2013)

Further preprocessing The fMRI scans from each dataset are further preprocessed to create DBGs. First, each scan is transformed into a multivariate timeseries of BOLD signals using the Glasser atlas (Glasser et al., 2016) to average voxels within $V = 360$ brain regions. Next, to ensure comparability with UKB, we truncate the length of HCP timeseries to 490 timepoints. Following the commonly used sliding-window method (Calhoun et al., 2014), we use Pearson correlation to calculate FC matrices within non-overlapping windows of length $1 < W \leq 490$ along the temporal dimension. At every window, we create an edge set of a unweighted and undirected graph with no self-edges by thresholding the top $1 \leq \epsilon < 100$ percentile values of the lower triangle of the FC matrix (excluding the principal diagonal) as connected following Kim et al. (2021). For both datasets, we choose $W = 30$ and $\epsilon = 5$ resulting in $T = \lfloor 490/30 \rfloor = 16$ graph snapshots each with $E^{(s,t)} = \lfloor (360(360 - 1)/2)(5/100) \rfloor = 3,231$ edges.

D. Baselines

We compare NeuroEvolve against a range of static and dynamic unsupervised probabilistic graph representation learning models, all with publicly available code. We leave comparisons to popular deterministic baselines such as Dynamic-Triad (Zhou et al., 2018), DySAT (Sankar et al., 2020), and DynNode2Vec (Mahdavi et al., 2018) for future work. Since all of the baselines were originally designed to model large single-subject dynamic graphs, we had to further adapt each implementation to work with multi-subject dynamic graphs.

Variational graph auto encoder⁴ (VGAE) (Kipf & Welling, 2016b) An extension of the variational autoencoder (Kingma & Welling, 2013) for graph structured data. Specifically, VGAE uses a graph convolutional network (GCN) (Kipf & Welling, 2016a) to learn a distribution over node embeddings. Originally designed for static graphs, we train VGAE on each dynamic graph snapshot independently.

Overlapping stochastic block model⁵ (OSBM) (Mehta et al., 2019) A deep generative version of the overlapping stochastic block model (Miller et al., 2009). In particular, OSBM places a stick-breaking prior over the number of communities which allows the model to automatically infer the optimal number of communities from the data during training. Similar to VGAE, OSBM uses a GCN to parameterize the distribution over node embeddings and is designed for static graphs.

Variational graph recurrent neural network⁶ (VGRNN) (Hajiramezanali et al., 2019) An extension of VGAE for dynamic graphs. Using a modified graph recurrent neural network, VGRNN is able to learn dependencies between and within

²<https://www.ukbiobank.ac.uk>

³<https://www.humanconnectome.org>

⁴<https://github.com/tkipf/gae>

⁵<https://github.com/nikhil-dce/SBM-meet-GNN>

⁶<https://github.com/VGraphRNN/VGRNN>

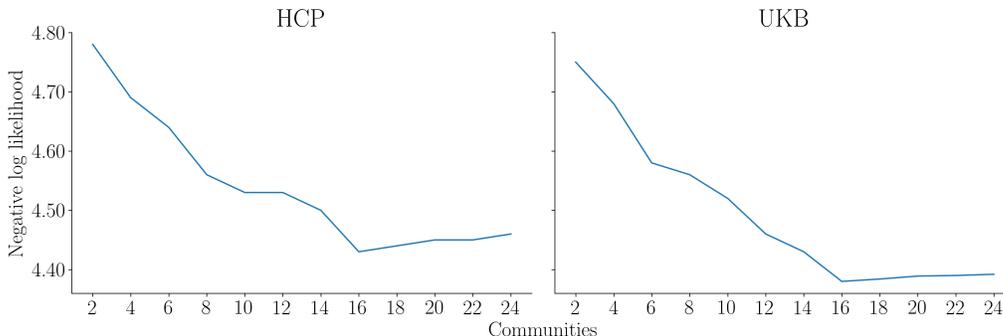


Figure 5. Elbow plot for finding the optimal number of communities K based on validation NLL.

changing graph topology over time. Similar to NeuroEvolve, the prior distribution over node embeddings is parameterized using embeddings from previous snapshots.

Evolving latent space model⁷ (ELSM) (Gupta et al., 2019) A generative model for dynamic graphs that learns node embeddings and performs community detection. In particular, node embeddings are initially sampled from a Gaussian mixture model over communities and then evolved over snapshots using long-short term memory units (Hochreiter & Schmidhuber, 1997). Unlike the previous baselines, ELSM does not use a GNNs to parameterize model distributions.

vGraph⁸ (VGRAPH) (Sun et al., 2019) Similar to NeuroEvolve, VGRAPH simultaneously learns node embeddings and community assignments by modeling nodes as being generated from a mixture of communities. The generative process of VGRAPH also relies on edge information. Since VGRAPH only models static graphs, we train it on each dynamic graph snapshot independently.

E. Implementation details

Software and hardware All models were developed in Python 3.7 (Python Core Team, 2019) using scikit-learn 1.1.1 (Pedregosa et al., 2011), PyTorch (Paszke et al., 2019), and numpy 1.1.1 (Harris et al., 2020). Statistical significance tests were carried out using deep-significance 1.1.1 (Ulmer et al., 2022). Experiments were performed on a Linux server (Debian 5.10.113-1) with a NVIDIA RTX A6000 GPU with 48 GB memory and 16 CPUs.

Hyperparameter optimization We use model and training hyperparameter values described in the original implementation of each baseline as a starting point for tuning on the validation dataset. Since searching for optional values for each hyperparameter configuration was outside the scope of the paper, we focused mainly on tuning the dimensions of NN hidden layers and embeddings, as well as the learning rate. For OSBM, VGRAPH and ELSM, we set the number of communities to the optimally tuned value of NeuroEvolve. To prevent overfitting, all models were trained using early-stopping with a patience of 15 based on the lowest validation NLL.

For NeuroEvolve, we choose the optimal number of communities $K = 16$ using the the elbo method (Thorndike, 1953) applied to validation NLL as shown in Figure 5. In the generative model, we fix the temporal smoothness hyperparameters $\sigma_\phi = \sigma_\psi = 0.01$. In the inference model, we fix the number of layers for all NNs to $L_\phi = L_\psi = L_z = 1$. For the Gumbel-softmax reparameterization trick we anneal the softmax temperature parameter starting from a maximum of 1 to a minimum of 0.05 at a rate of $3e-4$.

⁷<https://github.com/sh-gupta/ELSM>

⁸<https://github.com/fanyun-sun/vGraph>