

SEQUENTIAL MONTE CARLO GRAPH CONVOLUTIONAL NETWORK FOR DYNAMIC BRAIN CONNECTIVITY

Fengfan Zhao, Ercan Engin Kuruoglu

Tsinghua-Berkeley Shenzhen Institute, Shenzhen International Graduate School, Tsinghua University

ABSTRACT

An increasingly important brain function analysis modality is functional connectivity analysis which regards connections as statistical codependency between the signals of different brain regions. Graph-based analysis of brain connectivity provides a new way of exploring the association between brain functional deficits and the structural disruption related to brain disorders, but the current implementations have limited capability due to the assumptions of noise-free data and stationary graph topology. We propose a new methodology based on the particle filtering algorithm, with proven success in tracking problems, which estimates the hidden states of a dynamic graph with only partial and noisy observations, without the assumptions of stationarity on connectivity. We enrich the particle filtering state equation with a graph Neural Network called Sequential Monte Carlo Graph Convolutional Network (SMC-GCN), which due to the nonlinear regression capability, can limit spurious connections in the graph. Experiment studies demonstrate that SMC-GCN achieves the superior performance of several methods in brain disorder classification.

Index Terms— Brain Connectivity, Sequential Bayesian Learning, Sequential Monte Carlo, Particle Filtering, Graph Convolutional Network

1. INTRODUCTION

Functional dynamics include changes in the strength of connections between regions, and also the number of connections linked to regions. Various works [1, 2, 3] independently devised different types of state-space models to explore the dynamic characteristics of functional activation and applied them to event-related fMRI data analysis. Brain connectivity can be quantified by encoding neighbourhood relations into a connectivity matrix, the rows and columns of which correspond to different brain regions of interest (ROI). This representation lends itself to be mapped to a graphical model which provides means to quantify different topological aspects of the connectome. Recent years have witnessed an exponential growth of studies on the applications of Graph Convolutional Networks (GCNs) in neuroscience, in particular in finding common patterns or biomarkers [4].

The GCN-based methods on fMRI data can be categorized into two subgroups depending on the definition of nodes in the graph, i.e., population graph-based models and brain region graph-based models [5]. These two graph models correspond to two separate tasks for fMRI analysis, the node classification tasks and graph classification tasks. In population graph-based models, the nodes in the graph denote the subjects and the edges represent the similarity between subjects. PopulationGCN [6] involves representing populations as a sparse graph in which nodes are associated with imaging features and edge weights are constructed from phenotype information (e.g., age, gene, and sex of the subjects). One of the methods, BrainGNN [4], conducts an interpretable GCN model on a brain region graph to understand which brain regions are related to a specific neurological disorder.

Both of these graph-based models only used static functional connectivity information, where pairwise correlations between regions are calculated using the entire duration of the fMRI scan. This neglects the fact that brain connectivity is dynamic, with functional connectivity fluctuating over time. Dynamic connectivity presents time-varying region connections as opposed to static connectivity, which is becoming the frontier in fMRI data discovery [7].

A dynamic system estimation problem requires a dynamic model estimation method. Sequential Monte Carlo (SMC) provides a solution to dynamic system tracking problems by estimating the hidden states of a dynamic system with only partial and noisy observations and had important success in various applications [8, 9]. A specific SMC methodology called Particle Filter (PF) provides an extension to Kalman filtering to nonlinear systems and possibly non-Gaussian time-series and noise by employing a Monte Carlo sampling approach to iteratively track the posteriors of hidden variables of a system of interest using observations under noise. PF was applied to simulated recordings of electrical and neurovascular mediated hemodynamic activity, and the advantages of a unified framework were shown [10]. Ancherbak et al. implement PF application on gene network dynamics [11] and Ambrosi et al. [12] apply PF and vector autoregressive model in brain connectivity for dynamic modelling. Despite encouraging results obtained with these methods, the graph topological information is not utilised beyond a linear vector regression model for hidden variables.

In this work, we provide an extension to the nonlinear state evolution model while exploiting graph topological information by utilising a Graph Convolutional Network for uncovering connectomes from fMRI images of the brain. Our method, namely Sequential Monte Carlo Graph Convolutional Network (SMC-GCN), applies the particle filter to graphs with GCN backbone providing dynamic connectivity robustness and avoiding spurious links. The experiment result demonstrates that SMC-GCN achieves superior performance compared to other dynamic connectivity methods for brain disorder classification.

2. METHODOLOGY

2.1. Particle Filtering

Given the system model:

$$\begin{aligned} x_0 &\sim p_0(x) \\ x_t &= g_t(x_{t-1}, v_t), \text{ for } t \geq 1, \\ o_t &= h_t(x_t, w_t), \text{ for } t \geq 1 \end{aligned} \quad (1)$$

where v_t and w_t are the system transition and observation noise, g_t and h_t are functions of system. Consider a Markov process with the state transition probability given by $p(x_t|x_{t-1})$ where x_t is the state at the timestamp t . For every transition, we receive an observation o_t described by the observation probability of $p(o_t|x_t)$. After a sequential of observations $o_{t=1:T} = \{o_1, \dots, o_T\}$, particle filtering estimates the posterior probability $p(x_T|o_{t=1:T})$ of the state x_T conditioned on the sequence of observations. Particles are sequentially updated with three steps: First is the state transition step with the previous posterior state and transition model,

$$p(x_t|o_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|o_{1:t-1})dx_{t-1} \quad (2)$$

Then is the state update step with the current observation and transition state,

$$p(x_t|o_{1:t}) = \frac{p(o_t|x_t)p(x_t|o_{1:t-1})}{p(o_t|o_{1:t-1})} \quad (3)$$

as well as the weight update following the proposal density $q(\cdot)$ rather than the posterior observation $p(\cdot)$.

$$w_t^k \propto w_{t-1}^k \frac{p(o_t|x_t^k)p(x_t^k|x_{t-1}^k)}{q(x_t^k|x_{t-1}^k, o_t)} \quad (4)$$

The last is the resampling step. To avoid the degeneracy problem, particles need to resample K from the discrete distribution of $\langle w_{t+1}^k \rangle_{k=1:K}$ and reset weights to uniform distribution $1/K$.

2.2. Sequential Monte Carlo Graph Convolutional Network

Most of the existing work utilizes linear state updates for uniform or multivariate variables. In this work, we apply particle filtering on graphs where the observables are the node features while the state variable is the graph structure. Unlike classical particle filtering usage, we use a nonlinear model for state changes for which we utilize a Graph Convolutional Network. In this setup, since the state variables are graphs, the particles become graphs themselves to which weights are assigned. Rather than vectors, graphs are sampled forming *graph particles*. The overall architecture is shown in Fig. 1.

2.2.1. Graph Construction

For initialization, graph particles are sampled from the ROI time series with the length Γ timestamps. Each graph particle $\mathbf{X}_t^k = \{A_t^k, F_t^k\}$ is constructed by the node of ROIs defined by a specific atlas. The adjacency matrix A is computed by partial correlations between ROIs for the sparse weighted connection between nodes and the feature embeddings F corresponding to each ROIs are the correlation computation between ROIs for further transition and observation steps.

To construct a sequence of dynamic graph observations, we utilize the popular sliding window approach [13]. Given a window length Γ , and stride S , the $T = \lfloor T_{\max} - \Gamma/S \rfloor$ windows are constructed. Each observation $\mathbf{O}_t = \{A_t, F_t\}$ has the same construction with states.

2.2.2. State Transition

We make the GCN model for state transition, motivated by the Particle Filtering Network [14] replacing the observation model using CNN for visual localization. For the state transition step, each graph particle at timestamp $t-1$ will feed into the GCN backbone as the next state estimation where the system noise is simulated by different graph particles. It is the prior calculation for the next timestamp t state, the same as the GCN learning and prediction. The state transition equation is shown as follows:

$$\mathbf{X}_t^k | \mathbf{O}_{1:t-1}^k = \text{GCN}(\mathbf{X}_{t-1}^k | \mathbf{O}_{1:t-1}^k; \theta) \quad (5)$$

where θ is the learnable parameter in GCN with shared parameters during T timestamp transition and observation. Any specific convolution operation including Chebyshev polynomials [15] or simple GCN convolution [16] can be applied to the GCN model.

2.2.3. State Update

The state update step is the *posterior* calculation that the input graph particle at the timestamp t \mathbf{X}_t^k is updated by the current observation \mathbf{O}_t . Firstly, the adjacency matrices are aggregated by the top-K largest observation adjacency $A_t^k =$

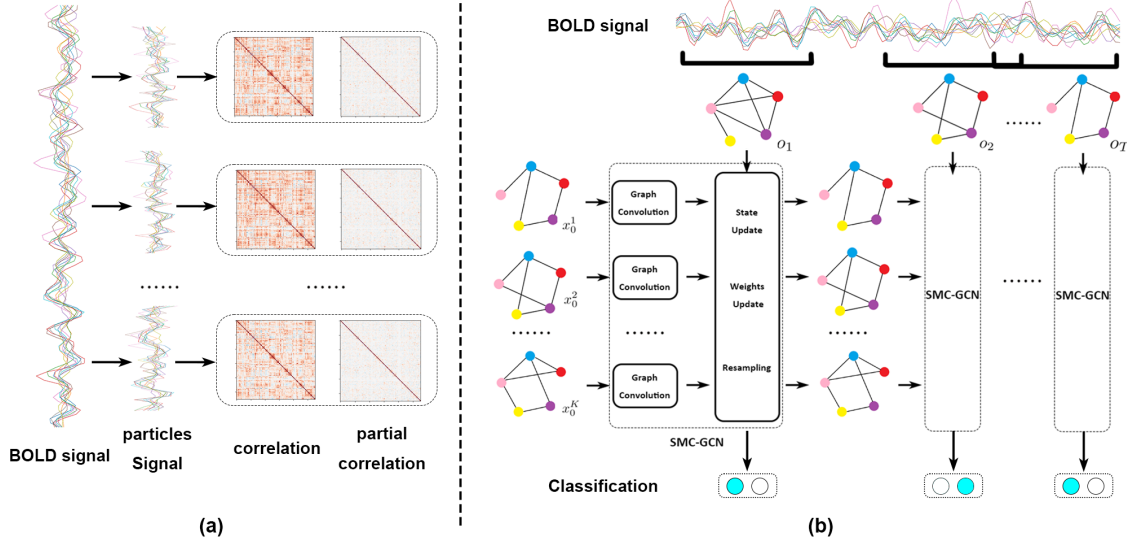


Fig. 1. Overall Architecture of SMC-GCN for Dynamic Brain Connectivity. (a) The fMRI time series is split into T time series by sliding window approach [13] for graph construction. After that, K graph particles $\{\mathbf{X}_t^k\}_{k=1:K}^{t=0:T}$ with initial particle weights $w^k = 1/K$ are constructed. (b) K graph particles are fed into SMC-GCN simultaneously for state transition acquiring priors for the next timestamp \mathbf{X}_{t+1}^k , following the state observation by \mathbf{O}_{t+1}^k and resampling over time. Each timestamp outputs the readout classification result.

$A_t^k(\mathbf{X}) + \text{TopK}(A_t(\mathbf{O}))$ and the feature matrices are modified to $F_t^k = F_t^k(\mathbf{X}) \circ F_t^k(\mathbf{O})$ with the Hadamard product. The state update aggregation is shown as follows:

$$\mathbf{X}_t | \mathbf{O}_{1:t} = \text{AGGREGATE}(\mathbf{O}_t, \mathbf{X}_t | \mathbf{O}_{1:t-1}) \quad (6)$$

In the absence of intermediate probability density for graph observation, we have used discriminative function $f(\mathbf{O}_t, \mathbf{X}_t^k)$ to up/down-weight the state. This function discriminates the differences in node features between observations and predicted states. First, it aggregates the neighbour features using K nearest neighbours and then makes the comparison with the predicted state.

$$f(\mathbf{O}_t, \mathbf{X}_t^k) = \exp \left\{ \sum_{v \in \mathcal{V}} \log p(\mathbf{O}_{t,v} | \mathbf{X}_t^k) \right\} \quad (7)$$

$\log p(\mathbf{O}_{t,v} | \mathbf{X}_t^k)$ denote the log-likelihood of the measurement at node v and time t . Since it is discriminatively trained to optimize the end task, it performs the same function as $p(o|x)$ in particle filtering.

$$w_t^k = \frac{w_{t-1}^k f(\mathbf{O}_t, \mathbf{X}_t^k)}{\sum_k w_{t-1}^k f(\mathbf{O}_t, \mathbf{X}_t^k)} \quad (8)$$

The $\sum_k w_{t-1}^k f(\mathbf{O}_t, \mathbf{X}_t^k)$ is the normalization term.

2.2.4. Soft Resampling

For the particle filtering resampling step, the weights are adjusted to the uniform distribution. Since this is a non-

differentiable operation, we adopt the differentiable soft-resampling strategy [14]. Specifically, we sample new particles from the convex combination of particle weights and uniform distribution,

$$w_t^{rk} = \frac{p_t(k)}{q_t(k)} = \frac{w_t^k}{\alpha w_t^k + (1-\alpha)1/K} \quad (9)$$

where $\alpha \in [0, 1]$ is a tunable parameter. This differentiable approximation provides non-zero gradients for the full particle chain with a trade-off between the desired sampling distribution ($\alpha = 1$) and the uniform sampling distribution ($\alpha = 0$).

2.3. Graph Particle Classification

Lastly, we seek a flattening operation to preserve information about the input graph in a fixed-size representation. Concretely, to summarize the output graph of the convolutional global pooling block, we use both mean and max pooling:

$$z_t^k = \text{mean}(\mathbf{H}_t^k) \parallel \text{max}(\mathbf{H}_t^k) \quad (10)$$

where H is the hidden embeddings during each transition GCN, with mean and max operating elementwisely, and \parallel denotes concatenation. To retain information on a graph in a vector, we concatenate both mean and max summarization for a more informative graph-level representation. The final prediction is submitted to an MLP to obtain the graph classification with particle weights as average results.

$$\hat{y}_{t,c} = \sum_{k=1}^K w^k \cdot \text{MLP}(z_t^k) \quad (11)$$

Table 1. Comparison of Cross-Validation Results on ABIDE Dataset

Method	Accuracy	Sensitivity	Specificity	AUC
SVM	61.87(\pm 0.36)	65.74(\pm 0.24)	69.54(\pm 0.26)	67.74(\pm 0.71)
RF	60.58(\pm 0.58)	52.30(\pm 0.91)	74.36(\pm0.83)	65.31(\pm 0.24)
GCN	68.28(\pm 0.84)	60.25(\pm 0.63)	71.54(\pm 0.46)	71.96(\pm 0.53)
GAT	66.24(\pm 0.73)	63.21(\pm 4.76)	69.45(\pm 0.78)	69.87(\pm 0.47)
BrainGNN	70.66(\pm 0.56)	67.30(\pm 0.99)	71.98(\pm 1.44)	72.71(\pm 0.87)
SMC-GCN	73.64(\pm0.98)	70.06(\pm1.56)	73.48(\pm 1.28)	75.29(\pm1.01)

Cross-entropy loss is used in the final classification for all timestamps T .

3. EXPERIMENTS

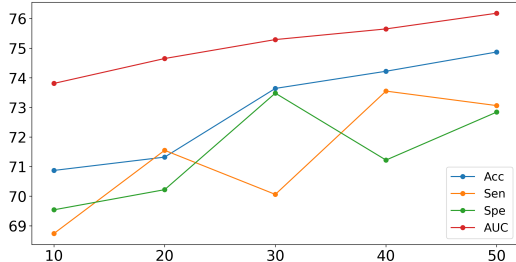
3.1. Dataset

We apply our model to the large and challenging database for binary classification tasks. The Autism Brain Imaging Data Exchange (ABIDE) database [17] aggregates data from different acquisition sites and openly shares functional MRI and phenotypic data of 1112 subjects from across 20 international sites. To ensure a fair comparison with the previous works, we use the same preprocessing pipeline, the Configurable Pipeline for the Analysis of Connectomes (C-PAC), with band-pass filtering and no global signal regression. We take the 871 subjects that passed manual quality control checks from three expert human reviewers and further prune the data with less than 160 time steps to a final total of 578 samples. We use the Harvard-Oxford (HO) atlas for ROI splitting. The resulting class distribution is 287 subjects with ASD to 291 healthy controls.

3.2. Experiment Setting

We utilize a stratified 5-fold cross-validated training procedure with a data split of 60% training, 20% validation, and 20% testing during hyperparameter tuning. The stratification occurs over both class (ASD and HC) and site (one of 17 scanning locations), in an attempt to minimize the effects of the different scanning parameters on our All training and testing are conducted on one NVIDIA RTX 3090 GPU. We use two layers of Chebyshev convolution with polynomials $K = 3$. We train our model using the Adam optimizer with the learning rate of 0.01 and the batch size is set to 12. The sliding window size is 100 and the stride is fixed for timestamp $T = 20$. The number of particles is $K = 30$.

The 5-fold cross-validation results of our methods are presented in Table 1, with classification performance compared to previous results. The baseline models consist of machine learning methods including SVM, Random Forest (RF) and graph based approaches including GCN, Graph Attention Networks (GAT) [18] and BrainGNN [4]. We achieve superior performance among baseline models.

**Fig. 2.** Number of Particles Performance

3.3. Ablation Study

We mainly focus on the number of particles ablation. For traditional particle filtering algorithms, particles sampled from pdf perform better when the particle number explodes. So for graph particles, we sample 10 to 50 particles in experiments due to training graph size. The results are shown in Table 2, with smooth growth of the accuracy and AUC score, while the training time is doubled, compared to number = 10. Since no thresholds appear when the particle number grows, we will further investigate the efficiency with the growth of the particle number for sampling approximation.

4. CONCLUSION

In this work, we have extended the classical particle filtering to a broader interpretation where the particles could correspond to graphs or other data embedding structures. We have further enriched the learning process with neural networks leading to the the Sequential Monte Carlo Graph Convolutional Network (SMC-GCN). This work has been motivated by the need to follow brain dynamics and has been used to model sequential connectivity in brain disorders. The method demonstrates superior performance compared to the baseline methods. Our future work will explore other applications of the SMC-GCN model with other dynamic graph topologies and its capacity for the growth of particles.

5. REFERENCES

- [1] Sylvain Faisan, Laurent Thoraval, J-P Armspach, and Fabrice Heitz, "Hidden markov multiple event sequence models: a paradigm for the spatio-temporal analysis of fMRI data," *Medical Image Analysis*, vol. 11, no. 1, pp. 1–20, 2007.
- [2] Rebecca A Hutchinson, Radu Stefan Niculescu, Timothy A Keller, Indrayana Rustandi, and Tom M Mitchell, "Modeling fMRI data generated by overlapping cognitive processes with unknown onsets using hidden process models," *NeuroImage*, vol. 46, no. 1, pp. 87–104, 2009.
- [3] Firdaus Janoos, Raghu Machiraju, Shantanu Singh, and Istvan Akos Morocz, "Spatio-temporal models of mental processes from fmri," *Neuroimage*, vol. 57, no. 2, pp. 362–377, 2011.
- [4] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H. Staib, Pamela Ventola, and James S. Duncan, "BrainGNN: Interpretable brain graph neural network for fmri analysis," *Medical Image Analysis*, vol. 74, pp. 102233, 2021.
- [5] Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekek, "Graph neural networks in network neuroscience," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2022.
- [6] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrero, Ben Glocker, and Daniel Rueckert, "Disease prediction using graph convolutional networks: Application to autism spectrum disorder and alzheimer's disease," *Medical Image Analysis*, vol. 48, pp. 117–130, 2018.
- [7] Maria Giulia Preti, Thomas AW Bolton, and Dimitri Van De Ville, "The dynamic functional connectome: State-of-the-art and perspectives," *NeuroImage*, vol. 160, pp. 41–54, 2017.
- [8] A. Doucet, A. Smith, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Information Science and Statistics. Springer New York, 2001.
- [9] Mauro Costagli and Ercan Engin Kuruoğlu, "Image separation using particle filters," *Digital Signal Processing*, vol. 17, no. 5, pp. 935–946, 2007.
- [10] Pierpaolo Croce, Filippo Zappasodi, Arcangelo Merla, and Antonio Maria Chiarelli, "Exploiting neurovascular coupling: a Bayesian sequential monte carlo approach applied to simulated eeg fnirs data," *Journal of Neural Engineering*, vol. 14, no. 4, pp. 046029, jul 2017.
- [11] Sergiy Ancherbak, Ercan E Kuruoglu, and Martin Vingron, "Time-dependent gene network modelling by sequential monte carlo," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 6, pp. 1183–1193, 2015.
- [12] Pierfrancesco Ambrosi, Mauro Costagli, Ercan E. Kuruoğlu, Laura Biagi, Guido Buonincontri, and Michela Tosetti, "Modeling brain connectivity dynamics in functional magnetic resonance imaging via particle filtering," *Brain Informatics*, vol. 8, no. 1, pp. 19, Sep 2021.
- [13] Fatemeh Mokhtari, Milad I Akhlaghi, Sean L Simpson, Guorong Wu, and Paul J Laurienti, "Sliding window correlation analysis: Modulating window shape for dynamic brain connectivity in resting state," *Neuroimage*, vol. 189, pp. 655–666, 2019.
- [14] Peter Karkus, David Hsu, and Wee Sun Lee, "Particle filter networks with application to visual localization," in *Conference on Robot Learning*. PMLR, 2018, pp. 169–178.
- [15] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [16] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li, "Simple and deep graph convolutional networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1725–1735.
- [17] Cameron Craddock, Yassine Benhajali, Chu Carlton, Chouinard Francois, Alan Evans, András Jakab, Budhachandra Khundrakpam, John Lewis, li Qingyang, Milham Michael, Yan Chaogan, and Pierre Bellec, "The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives," *Frontiers in Neuroinformatics*, vol. 7, 01 2013.
- [18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.