Supplementary

We first provide more details about how we construct the dataset and then elaborate all the modules of the GGN framework as well as the training and inference procedures.

Pre-processing of raw dataset. The raw dataset TUSZ v1.5.2 collects from the clinical scalp-EEG equipments recoding 22 channels signals from international 10-20 system. We select 20 channels of signals, details of the channel selection can be found in the additional supplementary materials. Suppose the time-series signal of channel *i* is denoted as $\hat{X}_i = \{\hat{x}_i^t\}_{t=0}^{T-1}$ with in total *T* time steps. Then we compute the Discrete Fourier Transform ($\mathcal{F}$) of the raw signal by Fast Fourier Transform: $x_i^k = \mathcal{F}(\hat{X}_i) = \sum_{t=0}^{T-1} \hat{x}_i^t \, e^{-\frac{2\pi i}{T}kt}$ with only real part left, and assemble final features by taking the *log* of the amplitude: $X_i = \{log(Amplitude(x_i^k))\}_{k=0}^{T-1}$. Then an input time-series sample is donoted as $\mathcal{X}_s = \{X_i\}_{i=1}^{N} \in \mathbb{R}^{N \times C_0 \times T}$, where the N is 20 channels, C_0 is the 200Hz, *T* is totoal time steps. For the raw topology construction, we have tried several approaches, such as correlation, distance, and functional relation of channels. We find the functional relation map which is clusterd into 6 groups based on the lobe position of cerebrum, achives better performance thant other methods. We use $A^{(0)}$ (a weighted adjacency matrix) to denote this initial raw topology that is constructed by a thresholded Gaussian kernel applied on the Euclidean distance of each two electrodes **错误!未找到引用源。**. The dataset can be denoted as $\{\mathcal{X}, \mathcal{Y}, A^{(0)}\}$, where the $\mathcal{X}$ consists of 3050 time-series samples and $\mathcal{Y}$ is the corresponding label set.

Optimization goal of GGN. The ultimate goal of GGN is to classify the seizure types through the EEG signals, this goal can be formulated by an optimization perspective. Given the dataset $\{\mathcal{X}, \mathcal{Y}, A^{(0)}\}$. The objective of our GGN can be denoted as follows:

$$arg \max_{\theta} \prod_{s} P_{\theta}\left(\mathcal{Y}_s \middle| \mathcal{X}_s, A^{(0)}\right) \tag{1}$$

where the θ denotes the learnable parameters of GGN model.

Interaction graph generator. To learn the dynamic mutual interactions of epileptic connectivities, the graph generator models the interactions in a probabilistic perspective. A connectivity is represented by a topological graph, that is denoted by a weighted adjacency matrix. The top of Fig. 5 depicts how the graph generator generate graphs. It takes a raw topology and temporal features as input, and then utilizes three independent GNNs to learn a set of parameters of a mixture Gaussian distribution. Since the epileptic networks varies over time and lead to disrruptions of the functional connectivities, using a mixture Gaussian distribution instead of a static or a deterministic connectivity can provide a powerful expressiveness for representing these dynamics. Nevertheless, this mixture distribution is unknown in prior and our goal is to approximate such a distribution.

Given the dataset $\{\mathcal{X}, \mathcal{Y}, A^{(0)}\}$, where $\mathcal{X}$ represents the input EEG signals, $\mathcal{Y}$ represents the labels, and $A^{(0)}$ is the initial weighed adjacency matrix which is constructed by a thresholded Gaussian kernel applied on the Euclidean distance of each two electrodes **错误!未找到引用源。**. Without considering of latent topologies, the maximize-likelihood optimization goal is as follows:

$$arg \max_{\theta} \prod_{s} P_{\theta}\left(\mathcal{Y}_s \middle| \mathcal{X}_s, A^{(0)}\right) \tag{1}$$

where the θ denotes the learnable parameters of the model. After we introduce a set of latent graphs $\{A^{(m)}\}_{m=1}^{M}$ by calculating its marginal distritbution, the previous goal is equivalent to:

$$arg\max_{\theta}\prod_{s}\sum_{m=1}^{M}P_{\theta}\big(\mathcal{Y}_s, A^{(m)}\big|\,,\mathcal{X}_s, A^{(0)}\big) \tag{2}$$

However, solving this complex optimization problem has two difficulties: 1) $A^{(m)}$ is not given, we don't know the prior distribution of $A^{(m)}$. 2) the $A^{(m)}$ are discrete variables, whereas, all the edges in $A^{(m)}$ are dependent random variables, leading to a highly complex combinatorial problem and the problem is not differentiable that can be optimized by stochasitic gradient descent (SGD) in a neural network.

To tackle these difficulties, we first leverage a variational inference technique to transform the optimization goal into:

$$argmax_{\theta,\omega}\sum_{s}\mathbb{E}_{Q_{\omega}}\left[log\frac{P_{\theta}\big(\mathcal{Y}_s, A^{(m)}\big|\,,\mathcal{X}_s, A^{(0)}\big)}{Q_{\omega}\big(A^{(m)}\big|\,\mathcal{X}_s, A^{(0)}\big)}\right] \tag{3}$$

This objective goal is also known as ELBO **错误!未找到引用源。**. There are two unknown distributions need to be learned, i.e., $Q_{\omega}\big(A^{(m)}\big|\,\mathcal{X}_s, A^{(0)}\big)$ and $P_{\theta}\big(\mathcal{Y}_s, A^{(m)}\big|\,,\mathcal{X}_s, A^{(0)}\big)$. In fact, the $Q_{\omega}\big(A^{(m)}\big|\,\mathcal{X}_s, A^{(0)}\big)$ could be any possible distribution families. To optimize such a distribution, we use different mixture Gaussian distritbutions parameterized by ω for modeling each node representation correspondingly, so that the possibility of the connection between two nodes can be measured by these two mixture distributions. The ω is learned by GNN-based models via SGD. The $P_{\theta}\big(\mathcal{Y}_s, A^{(m)}\big|\,,\mathcal{X}_s, A^{(0)}\big)$ is approximated by the other parts of GGN framework parameterized by θ.

To make the learning procedure be differentiable and sample the categorical adjacency matrix, we apply two reparameterization tricks. As illustrated in Fig 6.a, temporal features are fed into three GNN-based models to learn the parameters of the mixture Gaussian distribution $\mathcal{M} = \sum_{k=1}^{K}\pi_k N_k(\mu_k, \sigma_k)$ with K components for each node where the (π_k, μ_k, σ_k) are learned by a GNN-based model respectively. To approximate

$Q_{\omega}\left(A_{i,j}^{(m)} = 1\middle|\,\mathcal{X}_s, A^{(0)}\right)$, we first draw two samples S_i, S_j from $\mathcal{M}_i$, $\mathcal{M}_j$. by using two reparameterization

stricks:

$$O_{\pi} = one\_hot(\pi) = \mathrm{Gumb}(\pi, \tau) \tag{4}$$

$$S_i = O_{\pi}^{T}\mu + O_{\pi}^{T}\sigma \cdot n_i \tag{5}$$

where $\pi = [\pi_1, \dots, \pi_K]^T$, and $\mathrm{Gumb}(\cdot, \tau)$ is a Gumbel-softmax reparameterization trick **错误!未找到引用源。** to get a continuous one-hot representation O_{π} of a categorical sample drawn from the distribution π, and a temperature hyperparamer τ is to control the smoothness of O_{π}. Based on O_{π}, sampling from $\mathcal{M}_i$ could be differentiable by the second reparameterization trick shown in equation (4), wherer $\mu = [\mu_1, \dots, \mu_K]^T$, $\sigma = [\sigma_1, \dots, \sigma_K]^T$, and n_i is a sample drawn from a standard normal distribution. Repeated equation (4) and (5), we get the other sample S_j. Then the probability of the connection between node *i* and node *j* could be computed by:

$$Prob\left(A_{i,j}^{(m)} = 1\middle|S_i, S_j\right) = \mathrm{Sigmoid}\big(S_i \cdot S_j\big) \tag{6}$$

After calculated all probabilities of all node pairs, we obtain the $A_s^{(m)}$ for the input sample $\mathcal{X}_s$. Each sample in a training batch $\mathcal{B}$ corresponds to a new $A_s^{(m)}$ by repeating the equation (4)-(6). Here, we take the expectation of the $A_s^{(m)}$, i.e., $A\,\hat{}\,m = \mathbb{E}_{s\sim\mathcal{B}}\left[A_s^{(m)}\right]$.