

Supplementary

Pre-processing of raw dataset

EEG channel selection. The raw dataset TUSZ v1.5.2 [1] collects from the clinical scalp-EEG equipment, recording 22 channel signals from the international 10-20 system. We select 20 channels out of 22 and use TCP montage [2] as the time-series input signals, i.e., FP1-F7;F7-T3;T3-T5;T5-O1;FP2-F8;F8-T4;T4-T6;T6-O2;T3-C3;C3-CZ;CZ-C4;C4-T4;FP1-F3;F3-C3;C3-P3;P3-O1;FP2-F4;F4-C4;C4-P4;P4-O2.

Frequency feature extraction. 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 different frequencies in various bands of the raw signal in the one second time window 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 real part kept only, and take the \log of the amplitude: $X_i = \{\log(\text{Amplitude}(x_i^k))\}_{k=0}^{T-1}$. To this end, a time-series sample is denoted as $\mathcal{X}_s = \{X_i\}_{i=1}^N \in \mathbb{R}^{N \times C_0 \times T}$, where the N is 20 channels, $C_0 = 244$ is the total number of frequencies we used, T is total seizure attacking time in second.

Construction of raw topology. We use $A^{(0)}$ (a weighted adjacency matrix) to denote the initial raw topology that is constructed by a Gaussian kernel [2] with a threshold applied on the Euclidean distance of each two electrode. The weight value A_{ij} in $A^{(0)}$ between the node i and j is calculated by:

$$A_{ij} = \begin{cases} \exp\left(-\frac{|\text{dist}(i, j)|^2}{\sigma}\right) & \text{if } \text{dist}(i, j) < \gamma, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where the $\text{dist}(i, j)$ represents the Euclidean distance between node i and j , and σ, γ are two hyperparameters.

Derivation of the ELBO. We apply the *maximum likelihood estimation* (MLE) on the training dataset $\{\mathcal{X}, \mathcal{Y}, A^{(0)}\}$ with N samples, where the likelihood function \mathcal{L} is as follows:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y}, A^{(0)}) = \prod_{s=1}^N P_{\boldsymbol{\theta}}(\mathcal{Y}_s | \mathcal{X}_s, A^{(0)}) \quad (1)$$

where the $\boldsymbol{\theta}$ denotes the learnable parameters of GGN model $P_{\boldsymbol{\theta}}$. By introducing a set of latent connectivity graphs as the random variables $A^{(m)}$, the $P_{\boldsymbol{\theta}}(\mathcal{Y}_s | \mathcal{X}_s, A^{(0)})$ is the marginal distribution of $P_{\boldsymbol{\theta}}(\mathcal{Y}_s, A^{(m)} | \mathcal{X}_s, A^{(0)})$, so that \mathcal{L} equals to:

$$\mathcal{L}(\boldsymbol{\theta}; \{\mathcal{X}, \mathcal{Y}, A^{(0)}\}) = \prod_{s=1}^N \sum_{m=1}^M P_{\boldsymbol{\theta}}(\mathcal{Y}_s, A^{(m)} | \mathcal{X}_s, A^{(0)}) \quad (2)$$

Imposing the log likelihood and a variational inference on \mathcal{L} :

$$\begin{aligned}
& \log \mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y}, A^{(0)}) \\
&= \log \prod_{s=1}^N \sum_{m=1}^M P_{\boldsymbol{\theta}}(\mathcal{Y}_s, A^{(m)} | \mathcal{X}_s, A^{(0)}) \\
&= \sum_{s=1}^N \log \sum_{m=1}^M P_{\boldsymbol{\theta}}(\mathcal{Y}_s, A^{(m)} | \mathcal{X}_s, A^{(0)}) \frac{Q_{\boldsymbol{\omega}}(A^{(m)} | \mathcal{X}_s, A^{(0)})}{Q_{\boldsymbol{\omega}}(A^{(m)} | \mathcal{X}_s, A^{(0)})} \\
&= \sum_{s=1}^N \log \sum_{m=1}^M Q_{\boldsymbol{\omega}}(A^{(m)} | \mathcal{X}_s, A^{(0)}) \frac{P_{\boldsymbol{\theta}}(\mathcal{Y}_s, A^{(m)} | \mathcal{X}_s, A^{(0)})}{Q_{\boldsymbol{\omega}}(A^{(m)} | \mathcal{X}_s, A^{(0)})} \\
&\leq \sum_{s=1}^N \sum_{m=1}^M Q_{\boldsymbol{\omega}}(A^{(m)} | \mathcal{X}_s, A^{(0)}) \log \frac{P_{\boldsymbol{\theta}}(\mathcal{Y}_s, A^{(m)} | \mathcal{X}_s, A^{(0)})}{Q_{\boldsymbol{\omega}}(A^{(m)} | \mathcal{X}_s, A^{(0)})} \\
&= \text{ELBO}
\end{aligned}$$

Here, the $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ are the learnable neural network parameters of the GGN, specifically, $\boldsymbol{\omega}$ is attributed to the connectivity graph generator, and $\boldsymbol{\theta}$ is attributed to other modules of the GGN.

Gumbel Sampler. Define a 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 $Q_{\boldsymbol{\omega}}(A_{i,j}^{(m)} = 1 | \mathcal{X}_s, A^{(0)})$, we first draw two samples S_i, S_j from $\mathcal{M}_i, \mathcal{M}_j$. by using two reparameterization tricks:

$$O_{\pi} = \text{one_hot}(\pi) = \text{Gumb}(\pi, \tau) \quad (4)$$

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

where $\pi = [\pi_1, \dots, \pi_K]^T$, and $\text{Gumb}(\cdot, \tau)$ is a Gumbel-softmax reparameterization trick [3] to get a continuous one-hot representation O_{π} of a categorical sample drawn from the distribution π , and a temperature hyperparameter 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), where $\mu = [\mu_1, \dots, \mu_K]^T$, $\sigma = [\sigma_1, \dots, \sigma_K]^T$, 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:

$$\text{Prob}(A_{i,j}^{(m)} = 1 | S_i, S_j) = \text{Sigmoid}(S_i \cdot S_j) \quad (6)$$

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., $A^m = \mathbb{E}_{s \sim \mathcal{B}}[A_s^{(m)}]$.

Attention mechanism. Xxx.

References

- [1] Obeid, Iyad, and Joseph Picone. "The Temple University Hospital EEG Data Corpus." *Frontiers in Neuroscience* 10 (2016)
- [2] Shuman, David I., et al. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains." *IEEE Signal Processing Magazine* 30.3 (2013): 83-98.
- [3] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel softmax. In *International Conference on Learning Representations*, 2017.