# A Technical Appendix

## A.1 Configuration of the device operating environment

This section delineates the computational framework employed in conducting the experiments, encompassing both hardware and software components. The hardware configuration comprises a powerful NVIDIA GeForce RTX 3090, coupled with a high-performance NVIDIA GeForce RTX 3080 Ti GPU, alongside the computational prowess of an AMD Ryzen 9 5950X 16-Core Processor CPU. These components synergize seamlessly to create a formidable computing environment. At the helm of this infrastructure is the Ubuntu 20.04 LTS operating system, providing a robust foundation for our endeavors. Bolstered by 64GB of high-speed DDR4 RAM, the system delivers unparalleled responsiveness and efficiency. Facilitating the intricate operations of deep learning is the PyTorch framework, version 1.12.1. This version is augmented with GPU support, leveraging the computational might of CUDA version 11.4. This symbiotic fusion of hardware and software elements empowers our research with the necessary tools to achieve exemplary results.

## A.2 Data Description

### A.2.1 Benchmark Simulation Data

Smith dataset[1] has been widely used in the literature owing to its rich, realistic simulated fMRI data for a wide range of underlying networks. The simulated fMRI data is generated with dynamic changes in connection strength between brain regions. Moreover, these connection strengths undergo modulation over time by additional random processes.

In our experiments, we aim to test the different abilities of methods including performance on small samples and high-noise data (run on every single subject). So we utilize 9 kinds of typical simulation cases to test the performance of the MetaR-LEC algorithm. Specifically, sim1 is set to high noise. Sim2 adds more connections among brain regions. Sim3 and Sim4 have more brain regions and connections. Sim5 shares the external inputs of a network. Sim6 adds the global mean confound to all the brain regions. Sim7 is set to backward connections between brain regions. Sim8 increases the strength of connections and Sim9 is set to nonstationary connections. Each simulation case consists of 50 subjects, with each session lasting for 600 seconds, closely resembling real-world scenarios. The time repetition (TR) is set at 3.0

[1] https://www.fmrib.ox.ac.uk/datasets/netsim/index.html

seconds, resulting in a pre-processed time series length of 200 data points, which can be considered a relatively small sample size. The detailed description of the 9 data sets we used is shown in Table 1. In each dataset, we perform individual analyses on every subject and present the mean and standard deviation across all subjects in the experimental results section.

Table 1: Description of the benchmark simulation data.

| Dataset | No.node | No.arc | Samples | Noise(%) | Other factors |
|---|---|---|---|---|---|
| Sim1 | 5 | 5 | 200 | 1.0 | |
| Sim2 | 5 | 7 | 200 | 1.0 | more connections |
| sim3 | 10 | 11 | 200 | 1.0 | |
| sim4 | 15 | 19 | 200 | 1.0 | |
| Sim5 | 5 | 5 | 200 | 1.0 | shared inputs |
| Sim6 | 5 | 5 | 200 | 1.0 | global mean confound |
| Sim7 | 5 | 5 | 200 | 1.0 | backwards connections |
| Sim8 | 5 | 5 | 200 | 0.1 | stronger connections |
| Sim9 | 5 | 5 | 200 | 0.1 | nonstationary connections |

### A.2.2 Real fMRI Dataset

The real fMRI time-series dataset used in this paper is resting-state fMRI data, which can be downloaded at https://github.com/shahpreya/MTlnet. We consider the following seven regions of interests (ROIs) from the medial temporal lobe. The detailed information on ROIs is shown in Table 2.

Table 2: The ROIs of the real fMRI dataset.

| NO. | ROI | Detailed description |
|---|---|---|
| 1 | CA1 | Cornu Ammonis1 |
| 2 | CA23DG | Cornu Ammonis2,3 and Dentate Gyrus |
| 3 | SUB | Subiculum |
| 4 | ERC | Entorhinal Cortex |
| 5 | BA35 | Brodmann Areas 35 |
| 6 | BA36 | Brodmann Areas 36 |
| 7 | PHC | Parahippocampal Cortex |

## A.3 Evaluation Metrics

To evaluate the performance of effective connectivity learning methods, we make statistics of learned results from the following three most common graph metrics: Precision, Recall, SHD, and F1 score (F1):

$$Precision = \frac{CA}{TA}, \tag{1}$$

$$Recall = \frac{CA}{|\mathbb{G}|}, \tag{2}$$

$$SHD = EA + MA + RA \tag{3}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{4}$$

where extra arcs (EA), missing arcs (MA), reverse arcs (RA), correct arcs (CA), and total arcs(TA) are obtained from comparison of the learned results and ground-truth networks. And $\mathbb{G}$ is the arc set of ground-truth networks. $|\mathbb{G}|$ is the cardinal number of the set $\mathbb{G}$. In detail, Precision, Recall, and F1 are commonly used measurements for brain EC estimation, the value range of precision and F1 is from 0 to 1. SHD is the total number of arcs required to be changed to the ground-truth networks. If the value of SHD is equal to 0, that means all arcs the method learned are correct.

## A.4 Baseline Methods

To test and verify the competitiveness of the MetaRLEC, we compare MetaRLEC with the other 8 brain effective connectivity learning methods, some of them are classical machine learning methods, such as Patel (2006), pwLiNGAM (2010, Sometimes abbreviated in the main text as 'pw'), large-scale Granger causality (2017, Abbreviated in the main text as 'lsGC'), Two-Step method (2019). And some of them are state-of-the-art deep learning methods, which are abbreviated as, EC-RGAN (2021), RL-EC (2022), CR-VAE(2023), DiffAN(2023). The parameter settings of these baseline methods are shown in Table.3.

Table 3: Parameter settings of baseline methods.

| Methods | Parameter settings |
|---|---|
| Patel | threshold = 0.3 |
| pwLiNGAM | method = 1 |
| lsGC[3] | cmp = 5, ARorder = 2, normalize = 1 |
| Two-Step | $\lambda = 2$ |
| EC-RGAN | lr = 0.01, dlr = 0.01, l1 = 0.1, II = 3, nh = 100 |
| RL-EC | $\alpha = 1.0, \beta = 2.0, q_0 = 0.98, \rho = 0.2$ |
| CR-VAE | context=20, $\lambda = 0.1$, lr = 5e-2, nh = 64 |
| DiffAN | lr = 0.001, $\beta_{start}$= 0.0001, $\beta_{end}$ = 0.02, nh = 1024 |

## A.5 Model Configuration

In our framework, For the actor, we use an architecture consisting of an encoder and a decoder. The encoder consists of 3 Transformer encoder blocks and uses the ReLU activation function. Each block is composed of 8 attention heads with 1024 hidden nodes, and the input dimension is 100. The decoder is composed of Bi-LSTM with 2 layers, and the number of hidden nodes is 256. After Bi-LSTM is an attention layer to get the output. For the pointer network, its convolution kernel size is 1. For the critic, we use a multi-layer perceptron model containing three fully connected layers. Each layer uses a linear transformation with both input and hidden dimensions of 256. For the meta-critic network, we employ a multi-layer perceptron model containing three fully connected layers as well. The input dimension is 512, and the hidden dimension is 256, the output dimension is 1.

The input data for MetaECRL has a feature dimension size of 200, We sample one batch for 100 observed data points, and repeat sampling 64 times. So the shape of input data is 64 × n × 100, where n is the number of nodes, and the shape of the input to the pointer network is 64 × d × 256. To improve the training efficiency and accuracy, we use the Adam optimizer and set the learning rates for the actor to 1e-4, and for the critic and meta-critic to 1e-3, respectively. We train 500 epochs and record the graphs with the highest rewards in every 50 epochs, the final algorithm output is the graph with the highest reward score. Post-processing (thresholding) can help us more effectively observe the correct and sparse EC network and our sparse threshold is set at 0.3. For the results of other baseline methods, we use the same post-processing.

## A.6 Further findings from the "Experimental Results on Benchmark Simulated fMRI Dataset" Section

In this section, We give a detailed analysis of each simulated case as shown in table 4. Fig.1 shows 9 methods on Smith simulated dataset for each subject data F1 values. Sim1 simulates high noise fMRI data (1%). From the results, we can notice that deep learning methods perform worse than traditional machine learning algorithms. This may be due to the simplicity of the models in traditional machine learning methods, whereas deep learning methods are less effective because of the small sample of data available for training. Compared with these algorithms, MetaRLEC achieves significant advantages in small samples and noisy data environments.

Sim2 adds two connections among brain regions. The results show that The performance of all methods does not differ much from the results of sim1, indicating that the increase of edges has a small effect on the performance of the algorithm. It is noted that the CR-VAE method achieves significant performance than in Sim1, we find that CR-VAE estimates more reverse edge so both recall and F1 are improved considerably. Sim3 and Sim4 add more nodes and edges to Sim1. From the results of Sim3 and Sim4, traditional machine learning methods suffer the most significant performance degradation due to constraints in their learning mechanisms, while deep learning methods, which excel at handling complex variable relationships, exhibit comparatively better performance. Our method also suffered a performance drop, but benefited from the advantages of the meta-critic framework, which is better able to discover causal relationships between brain regions from small samples of fMRI data. Nevertheless, MetaRLEC continued to demonstrate optimal or suboptimal performance across all methods.

Sim5 has the same ground truth as Sim1, but the external neuronal noises are mixed into the nodes. From the results of Sim5, we can notice that traditional machine learning algorithms are more affected by external noise inputs, while deep learning methods are less affected. Moreover, compared with other algorithms, MetaRLEC has a decrease in performance, but still maintains its relative advantages over other algorithms.

Sim6 includes global mean confounding factors into the fMRI time series of brain regions. The results of Sim6 indicate that most algorithms are robust to global confounding factors. Nevertheless, MetaRLEC continued to demonstrate superior performance across all evaluation metrics.

Sim7 reverses connections in ground truth networks. The results show that backward connections have less effect on most methods, The performance of this method is second only to the Two-Step method.

Table 4: The mean and the standard deviation results of 9 methods on Smith simulated dataset using single subject data.

| Data | Metrics | Methods | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Patel | pwLiNGAM | lsGC | Two-Step | EC-RGAN | RL-EC | CR-VAE | DiffAN | MetaRLEC |
| Sim1 | Precision | 0.40±0.09 | 0.34±0.10 | 0.36±0.20 | 0.56±0.16 | 0.32±0.11 | 0.35±0.36 | 0.22±0.09 | 0.40±0.25 | **0.62±0.16** |
| | Recall | 0.60±0.20 | **0.77±0.21** | 0.37±0.21 | 0.73±0.23 | 0.60±0.27 | 0.30±0.25 | 0.41±0.22 | 0.35±0.18 | 0.65±0.19 |
| | SHD | 2.92±1.30 | 5.35±1.30 | 4.94±1.65 | **1.60±1.27** | 4.86±1.41 | 3.58±1.30 | 6.32±1.26 | 3.40±1.00 | 2.28±1.03 |
| | F1 | 0.46±0.10 | 0.47±0.13 | 0.35±0.17 | 0.62±0.17 | 0.40±0.15 | 0.35±0.28 | 0.28±0.12 | 0.40±0.20 | **0.63±0.17** |
| Sim2 | Precision | 0.36±0.08 | 0.24±0.14 | 0.23±0.18 | 0.40±0.18 | 0.41±0.11 | 0.44±0.30 | 0.26±0.12 | 0.46±0.28 | **0.59±0.24** |
| | Recall | 0.57±0.19 | 0.63±0.21 | 0.37±0.21 | 0.50±0.21 | 0.48±0.25 | 0.22±0.16 | **0.73±0.18** | 0.29±0.18 | 0.55±0.28 |
| | SHD | 4.04±1.54 | 5.29±1.74 | 5.88±1.44 | 3.74±1.56 | 5.20±1.39 | 5.58±1.29 | 4.68±1.24 | 5.30±1.55 | **3.70±2.13** |
| | F1 | 0.46±0.10 | 0.45±0.14 | 0.35±0.17 | 0.49±0.17 | 0.41±0.17 | 0.29±0.21 | 0.45±0.09 | 0.35±0.22 | **0.57±0.26** |
| Sim3 | Precision | 0.28±0.08 | 0.18±0.03 | 0.24±0.11 | 0.44±0.11 | 0.14±0.02 | 0.50±0.10 | 0.12±0.02 | 0.52±0.18 | **0.54±0.14** |
| | Recall | 0.51±0.16 | 0.72±0.13 | 0.34±0.16 | 0.76±0.15 | **0.82±0.13** | 0.37±0.12 | 0.80±0.15 | 0.36±0.12 | 0.58±0.16 |
| | SHD | 10.38±2.76 | 28.49±3.43 | 14.78±3.17 | 6.62±3.06 | 32.44±2.21 | 7.02±1.40 | 33.74±1.92 | 7.40±1.51 | **6.32±2.20** |
| | F1 | 0.35±0.07 | 0.29±0.05 | 0.27±0.10 | 0.55±0.12 | 0.24±0.03 | 0.42±0.11 | 0.21±0.03 | 0.42±0.14 | **0.56±0.15** |
| Sim4 | Precision | 0.23±0.07 | 0.17±0.03 | 0.21±0.08 | 0.41±0.07 | 0.11±0.01 | 0.5±0.19 | 0.09±0.01 | 0.49±0.14 | **0.51±0.14** |
| | Recall | 0.40±0.11 | 0.74±0.10 | 0.29±0.12 | 0.68±0.12 | **0.91±0.05** | 0.22±0.11 | 0.80±0.10 | 0.31±0.10 | 0.51±0.14 |
| | SHD | 21.31±4.50 | 58.1±7.53 | 26.9±5.66 | 14.58±3.67 | 79.86±3.07 | 15.12±2.13 | 86.04±2.59 | 14.34±2.01 | **13.44±3.41** |
| | F1 | 0.28±0.06 | 0.27±0.04 | 0.23±0.07 | **0.51±0.08** | 0.2±0.01 | 0.30±0.13 | 0.16±0.02 | 0.38±0.11 | 0.51±0.14 |
| Sim5 | Precision | 0.34±0.09 | 0.29±0.07 | 0.32±0.20 | 0.44±0.15 | 0.32±0.11 | **0.5±0.3** | 0.26±0.14 | 0.44±0.22 | 0.49±0.22 |
| | Recall | 0.53±0.21 | **0.64±0.18** | 0.34±0.22 | 0.52±0.18 | 0.60±0.27 | 0.38±0.24 | 0.48±0.22 | 0.40±0.19 | 0.48±0.18 |
| | SHD | 3.77±1.31 | 5.86±1.26 | 5.14±1.45 | **3.24±1.23** | 4.86±1.41 | 3.70±1.44 | 6.10±1.40 | 3.98±1.29 | 3.46±1.44 |
| | F1 | 0.39±0.11 | 0.40±0.10 | 0.31±0.17 | 0.47±0.15 | 0.40±0.15 | 0.42±0.25 | 0.32±0.14 | 0.41±0.19 | **0.48±0.19** |
| Sim6 | Precision | 0.33±0.09 | 0.26±0.08 | 0.28±0.15 | 0.42±0.09 | **0.67±0.14** | 0.46±0.29 | 0.24±0.10 | 0.49±0.22 | 0.60±0.21 |
| | Recall | 0.62±0.21 | 0.65±0.19 | 0.39±0.21 | 0.81±0.18 | 0.66±0.26 | 0.42±0.27 | 0.41±0.21 | 0.49±0.21 | **0.68±0.25** |
| | SHD | 3.70±1.56 | 6.30±1.13 | 5.66±1.61 | **2.22±1.27** | 5.68±1.35 | 3.36±1.59 | 6.14±1.46 | 3.22±1.20 | 2.40±1.33 |
| | F1 | 0.42±0.10 | 0.37±0.11 | 0.32±0.15 | 0.55±0.11 | 0.63±0.16 | 0.43±0.27 | 0.29±0.13 | 0.48±0.20 | **0.64±0.22** |
| Sim7 | Precision | 0.35±0.09 | 0.32±0.08 | 0.32±0.21 | 0.62±0.21 | 0.37±0.17 | 0.43±0.27 | 0.25±0.05 | 0.48±0.20 | **0.64±0.22** |
| | Recall | 0.59±0.20 | **0.81±0.20** | 0.40±0.25 | 0.70±0.24 | 0.59±0.24 | 0.24±0.12 | 0.77±0.21 | 0.38±0.17 | 0.60±0.19 |
| | SHD | 3.54±1.26 | 5.51±1.08 | 5.28±1.76 | **1.64±1.29** | 4.48±1.13 | 5.40±0.92 | 5.76±1.14 | 3.26±0.92 | 2.82±1.00 |
| | F1 | 0.42±0.10 | 0.45±0.11 | 0.34±0.19 | 0.64±0.19 | 0.41±0.11 | 0.54±0.15 | 0.38±0.08 | 0.52±0.18 | **0.55±0.16** |
| Sim8 | Precision | 0.27±0.06 | 0.20±0.11 | 0.23±0.20 | 0.37±0.16 | 0.33±0.10 | **0.59±0.23** | 0.20±0.10 | 0.42±0.17 | 0.57±0.23 |
| | Recall | 0.58±0.17 | 0.34±0.23 | 0.25±0.22 | 0.44±0.23 | 0.58±0.22 | 0.62±0.21 | 0.39±0.20 | 0.52±0.19 | **0.7±0.25** |
| | SHD | 4.77±1.35 | 6.82±1.27 | 5.90±1.38 | 3.16±1.28 | 4.30±1.38 | **2.66±1.39** | 6.62±1.37 | 3.98±1.39 | 2.94±1.80 |
| | F1 | 0.36±0.08 | 0.24±0.14 | 0.23±0.18 | 0.40±0.18 | 0.41±0.11 | 0.60±0.21 | 0.26±0.12 | 0.46±0.17 | **0.63±0.24** |
| Sim9 | Precision | 0.35±0.09 | 0.16±0.12 | 0.27±0.21 | 0.47±0.26 | 0.28±0.05 | 0.17±0.07 | 0.45±0.29 | 0.49±0.28 | **0.53±0.17** |
| | Recall | 0.50±0.20 | 0.29±0.21 | 0.33±0.26 | 0.37±0.21 | **0.91±0.19** | 0.40±0.24 | 0.25±0.14 | 0.37±0.22 | 0.57±0.17 |
| | SHD | 3.67±1.19 | 7.45±1.67 | 5.36±1.33 | **3.36±1.02** | 5.00±0.81 | 7.52±1.15 | 4.84±0.96 | 3.74±1.19 | 3.7±1.26 |
| | F1 | 0.40±0.11 | 0.20±0.14 | 0.28±0.20 | 0.40±0.21 | 0.42±0.07 | 0.24±0.11 | 0.31±0.16 | 0.41±0.22 | **0.54±0.15** |

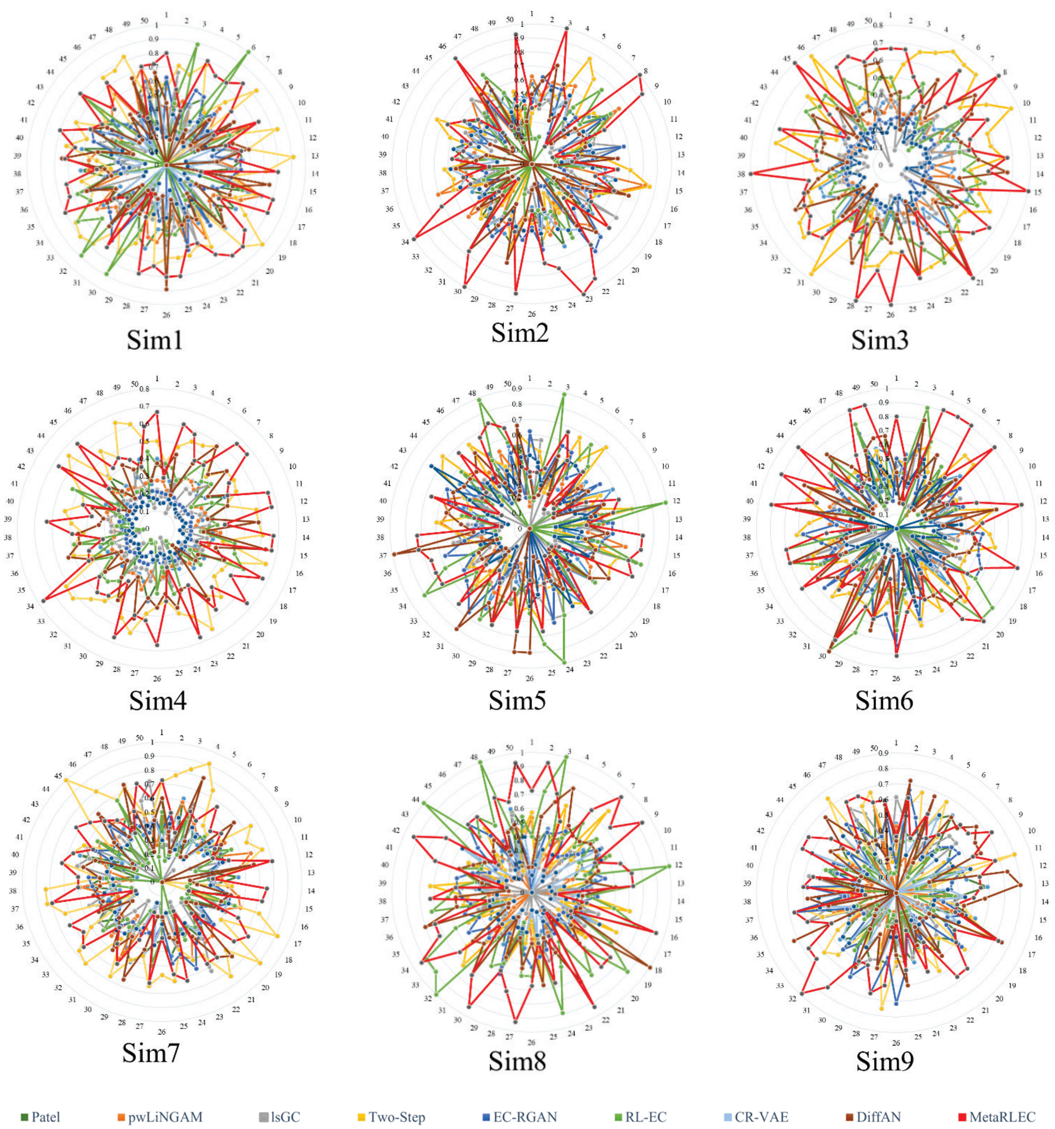


Figure 1: Results (F1 values) in terms of 9 methods on Smith simulated data set of 50 subjects. The outer ring is the serial number of subjects, and the inner ring shows the value of F1 which is from 0 to 1.

In Sim8, the connection strength is increased from 0.4 to 0.9. Results from Sim8 indicate that almost all algorithms performed worse than those of Sim1 to varying degrees. The performance degradation of the Twe-Step method is particularly significant, while the RL-EC method improves more significantly, indicating that reinforcement learning is robust in the face of changing environments. Beside, MetaRLEC still remains superior performance on recall and F1, which means that MetaRLEC has a strong robustness to connection strength change.

Sim9 investigates the effect of non-stationary connection strength between brain regions. The results of Sim9 indicate that traditional machine learning methods have a serious decline in evaluation metrics, while deep learning methods are more capable of handling non-stationary information. Moreover, MetaRLEC is obviously better at handling non-stationary information, thus achieving more outstanding metrics.

Overall, the findings suggest that MetaRLEC have significant potential for applications in small-sample fMRI data analysis and may provide more reliable and accurate insights into brain effective connectivity. Two-Step method thanks to its two-step design of learning the Structure first and then the direction, has achieved results second only to MetaRLEC.

To clearly show the statistical differences between these algorithms, we use the Friedman test and T-test to attest to the significant difference between these algorithms. In detail, we first perform Friedman-test on four metrics (precision, recall, F1, SHD) To further illustrate the differences between MetaRLEC and other comparison methods, we do the T-test on the results of MetaRLEC and other methods on eight datasets for four metrics, if the p-value is less than 0.05, then we consider there is a significant difference between MetaRLEC and the corresponding methods.

From Fig. 1 and Table 2 in the main text, we find that the precision of MetaRLEC is significantly different from all other methods. which indicates that MetaRLEC methods have a better performance than those methods in precision. Likewise, the test result on recall is significant. Two-step, EC-GAN, CR-VAE, and MetaRLEC learn most of the correct edges. Besides, we can see that there are significant differences between MetaRLEC and other methods on F1 as all p-value is less than 0.05 (except Two-Step method), which shows that MetaRLEC performs better than other baseline methods in this situation. Overall, the statistical test results show that compared with the state-of-the-art methods, our proposed method MetaRLEC has significant advantages in most evaluation metrics of most data sets.

From these experimental results, we can conclude that MetaRLEC performs better than the other 8 methods on the simulation datasets which has a significant difference in most of the performances compared to other methods.