Supplementary Material

CONTENTS

In the first section of this Supplementary Material, we give the python code of Algorithm 1. For the subtraction part in the code, we provide the theoretical explanation of why we need to use a subtraction between two `copent` functions (note the function already gets the negative value). We also give an example of using our algorithm. We get the adjacent matrix using the method in [Shimizu et al., 2011].

In the second section, we give more comparisons when the latent confounders are normal and exponential distributions, even though we can choose arbitrary distributions. In addition, we give the running time comparisons and how to select the hyperparameter in the algorithm.

A ALGORITHM 1

The main code of the LiNGAM-MMI (Algorithm 1) is as follows.

```
import numpy as np
import copent.copent as copent
def LiNGAM_MINE6(mat, k=10):
    class Node:
        def __init__(self , path , score , mat):
            self.path=path
            self.score=score
            self.mat=mat
    n , p = mat.shape
    times = 0
    node=Node([], 0.0, mat)
    OPEN=[node]
    while 1:
        min_score=float(’inf’)
        for node in OPEN:
            # print("1:",node.score)
            if node.score < min_score:
                min_score=node.score
                candidate=node
        node=candidate
        OPEN.remove(candidate)
        for j in [k for k in range(p) if k not in node.path]:
            path=node.path+[j]
            if len(path)==p:
                return path , times
            times=times+1
            flag=0
            for existing_node in OPEN:
                if set(path)==set(existing_node.path):
                    flag=1
                    conflict=existing_node

            index_set=[k for k in range(p) if k not in path]
            if(flag==0):
                new_mat=np.zeros([n,p])
                for i in index_set:
                    coeff=sum(node.mat[:,i]*node.mat[:,j])/sum(node.mat[:,j]**2)
                    new_mat[:,i]=node.mat[:,i] - coeff * node.mat[:,j]
                    # print(new_mat[:,index_set].shape,node.mat[:,j:(j+1)].shape)
                score=node.score+copent(np.concatenate([node.mat[:,j:(j+1)],
                      new_mat[:,index_set]],1),k=k)-copent(new_mat[:,index_set],k=k)
                # print(score)
                new_node=Node(path, score, new_mat)
                OPEN.append(new_node)
            else:
                score=node.score+copent(np.concatenate([node.mat[:,j:(j+1)],
                      conflict.mat[:,index_set]],1),k=k)-copent(conflict.mat[:,index_set],
                                                                   k=k)
                if score < conflict.score:
                    conflict.path=path
                    conflict.score=score
```

the next is an example of using the algorithm:

```
n = 500
x0 = np.random.laplace(0,1/np.sqrt(2),size=n)
f1 = 2 * np.random.uniform(size=n) - 1
x1 = x0 + np.random.laplace(0,1/np.sqrt(2),size=n) + f1
x2 = x1 + np.random.laplace(0,1/np.sqrt(2),size=n) + f1
x3 = x2 + np.random.laplace(0,1/np.sqrt(2),size=n)
x4 = x3 + np.random.laplace(0,1/np.sqrt(2),size=n)
x=np.array([x0, x1, x2, x3, x4]).T

xx = LiNGAM-MMI(x, k=10)
print("causal_order:", xx[0])
```

A.1 MAKING ADJACENT MATRIX FROM CAUSAL ORDER

To make the DAG, we use the function `_estimate_adjacency_matrix(x, _causal_order=xx[0])` in python package `lingam`, which constructs a strictly lower triangular matrix by following the order `xx[0]`, and estimate the connection strengths by using some conventional covariance-based regression such as least squares and maximum likelihood approaches on the original random vector and the original data matrix X. We use least squares regression in this paper as [Shimizu et al., 2011].

A.2 PROOF OF THEOREM 2

Algorithm 1 closes TOP and opens p nodes at the initial stage, each containing $p-1$ variables. Then, from the assumption, for large n, there is only one path with zero mutual information. In each step, Algorithm 1 closes one of the nodes that opened the latest at each step and outputs SHORTEST_PATH in p steps. The number of opened nodes is $p+(p-1)+\cdots+1$. This completes the proof.

Algorithm 1. Let $OPEN := \{TOP\}$, CLOSE:={}, $\mathrm{path}(\mathrm{TOP}) := ()$, $r(\mathrm{TOP})$:=DATA, and repeat:

```
1. Move the node $v \in$ OPEN to CLOSE s.t. $d(v)$ be min in OPEN and let $v_1, \cdots, v_m$ are connected to $v$;
2. For each $i = 1, \cdots, m$:
    (a) If $v_i \notin$ OPEN, compute the residue $r(v_i)$ of $v_i$ from $r(v)$;
    (b) Compute the MI $mi$ via $r(v)$ and $r(v_i)$.
    (c) If either $v_i \notin$ OPEN or $\{v_i \in$ OPEN, and $d(v) + mi < d(v_i)\}$, then $d(v_i) = d(v) + mi$ and $\mathrm{path}(v_i) = append(\mathrm{path}(v), v_i)$
    (d) join $v_i$ to OPEN if $v_i \notin$ OPEN for $j = 1, \ldots, m$.
3. If BOTTOM $\in$ OPEN, SHORTEST_PATH = $append(\mathrm{path}(v), \{\})$ and terminate.
```

A.3 HOW TO CALCULATE THE MUTUAL INFORMATION $I(x_1, \{x_2, \ldots, x_p\})$

Definition 1 (Copula Entropy). *Let $x \in \mathbb{R}^p$ be random variables with marginal functions $u = [F_1, \ldots, F_p]$ and copula density $c(u)$. The copula entropy of x is defined as*

$$H_c(x) = -\int_u c(u) \log(u) du \tag{1}$$

where $c(u) = \frac{d^p C(u)}{du_1 du_2 \ldots du_p}$.

Proposition 1 ([Ma and Sun, 2011]). *The mutual information of random variables is equivalent to the negative copula entropy:*

$$I(x) = -H_c(x). \tag{2}$$

Corollary 1. *Let $x = \{x_1, \ldots, x_p\}$ be a multivariate random variable that has p dimensions. The mutual information of $I\left(x_1, \{x_2, \ldots, x_p\}\right)$ is*

$$I\left(x_1, \{x_2, \ldots, x_p\}\right) = H_c\left(\{x_2, \ldots, x_p\}\right) - H_c(x) \tag{3}$$

Proof. From the property of multivariate mutual information, we know

$$I\{x_1, \ldots, x_p\} = I\{x_1, \{x_2 \ldots, x_p\}\} + I\{x_2, \ldots, x_p\}\ , \tag{4}$$

then

$$I\{x_1, \{x_2 \ldots, x_p\}\} = I\{x_1, \ldots, x_p\} - I\{x_2, \ldots, x_p\} \tag{5}$$

from Proposition 1, we know

$$I\{x_1, \{x_2 \ldots, x_p\}\} = H_c\{x_2, \ldots, x_p\} - H_c\{x_1, \ldots, x_p\}\ . \tag{6}$$

□

Table 1: Fifty trials for eight methods with $p = 10$ when no confounder exists, "Direct": DirectLiNGAM (pairwise), "DirectKernel": DirectLiNGAM (kernel-based), "ICA": ICA-LiNGAM, "MMI": our LiNGAM-MMI.

Data size	Metrics	Methods with **No Confounder** (dim=10, mean ± standard deviation)							
		Direct	DirectKernel	ICA	RCD	CAMUV	PC	RESIT	MMI (ours)
$n = 100$	Precision	0.73 ± 0.22	0.56 ± 0.11	0.70 ± 0.23	NA	0.49 ± 0.17	0.29 ± 0.09	0.14 ± 0.11	**0.96 ± 0.10**
	Recall	0.70 ± 0.29	0.73 ± 0.15	0.72 ± 0.23	0.02 ± 0.05	0.29 ± 0.14	0.31 ± 0.13	0.10 ± 0.08	**0.97 ± 0.09**
	F1	0.70 ± 0.25	0.63 ± 0.12	0.71 ± 0.23	NA	0.35 ± 0.14	0.30 ± 0.11	NA	**0.97 ± 0.10**
	SHD	5.68 ± 4.56	8.90 ± 2.78	5.88 ± 4.84	13.14 ± 0.82	11.56 ± 2.26	12.66 ± 2.17	18.68 ± 2.00	**0.70 ± 2.06**
$n = 200$	Precision	0.85 ± 0.13	0.58 ± 0.07	0.94 ± 0.13	NA	0.67 ± 0.20	0.26 ± 0.09	0.15 ± 0.11	**0.99 ± 0.04**
	Recall	0.97 ± 0.12	0.82 ± 0.09	0.96 ± 0.11	0.07 ± 0.10	0.61 ± 0.24	0.28 ± 0.11	0.10 ± 0.08	**1.00 ± 0.03**
	F1	0.91 ± 0.13	0.68 ± 0.07	0.95 ± 0.12	NA	0.63 ± 0.22	0.27 ± 0.09	NA	**0.99 ± 0.04**
	SHD	1.48 ± 3.34	7.88 ± 1.62	0.98 ± 2.49	12.80 ± 1.13	7.56 ± 4.54	12.62 ± 1.78	18.36 ± 2.11	**0.10 ± 0.57**
$n = 500$	Precision	**1.00 ± 0.00**	0.59 ± 0.06	**1.00 ± 0.00**	NA	0.92 ± 0.09	0.16 ± 0.08	0.26 ± 0.16	**1.00 ± 0.00**
	Recall	**1.00 ± 0.00**	0.83 ± 0.06	**1.00 ± 0.00**	0.30 ± 0.26	0.97 ± 0.06	0.18 ± 0.10	0.18 ± 0.11	**1.00 ± 0.00**
	F1	**1.00 ± 0.00**	0.69 ± 0.06	**1.00 ± 0.00**	NA	0.94 ± 0.07	0.17 ± 0.09	NA	**1.00 ± 0.00**
	SHD	**0.00 ± 0.00**	7.78 ± 1.36	**0.00 ± 0.00**	11.72 ± 2.07	1.52 ± 2.05	13.96 ± 1.74	16.66 ± 2.79	**0.00 ± 0.00**

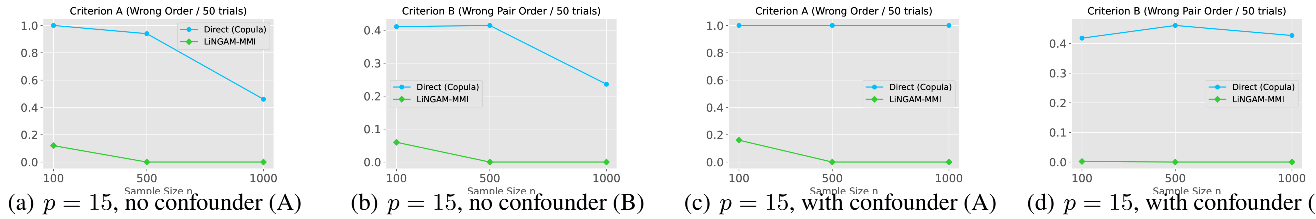
Data size	Metrics	Methods with **One Confounder** on $\{x1, x2\}$ (dim=10, mean ± standard deviation)							
		Direct	DirectKernel	ICA	RCD	CAMUV	PC	Parce	MMI (ours)
$n = 100$	Precision	0.69 ± 0.20	0.51 ± 0.10	0.68 ± 0.19	NA	0.41 ± 0.22	0.31 ± 0.10	0.38 ± 0.12	**0.97 ± 0.08**
	Recall	0.64 ± 0.24	0.67 ± 0.13	0.70 ± 0.20	0.00 ± 0.01	0.22 ± 0.13	0.32 ± 0.11	0.27 ± 0.09	**0.99 ± 0.07**
	F1	0.65 ± 0.21	0.58 ± 0.11	0.69 ± 0.19	NA	NA	0.32 ± 0.10	0.31 ± 0.09	**0.98 ± 0.07**
	SHD	6.56 ± 3.78	10.02 ± 2.78	6.56 ± 4.08	12.96 ± 0.20	12.52 ± 2.44	12.16 ± 2.07	12.66 ± 2.02	**0.47 ± 1.66**
$n = 200$	Precision	0.85 ± 0.19	0.54 ± 0.05	0.89 ± 0.16	NA	0.57 ± 0.20	0.25 ± 0.10	0.43 ± 0.12	**0.98 ± 0.08**
	Recall	0.84 ± 0.26	0.77 ± 0.08	0.91 ± 0.17	0.02 ± 0.03	0.48 ± 0.21	0.27 ± 0.12	0.34 ± 0.12	**0.99 ± 0.09**
	F1	0.84 ± 0.23	0.52 ± 0.19	0.90 ± 0.16	NA	0.63 ± 0.22	0.26 ± 0.11	0.37 ± 0.11	**0.98 ± 0.08**
	SHD	3.04 ± 3.96	8.88 ± 1.42	2.02 ± 3.31	12.96 ± 0.72	9.72 ± 3.48	13.12 ± 2.04	12.04 ± 1.84	**0.38 ± 1.59**
$n = 500$	Precision	0.98 ± 0.04	0.56 ± 0.05	0.98 ± 0.05	NA	0.62 ± 0.25	0.19 ± 0.07	0.47 ± 0.17	**0.99 ± 0.03**
	Recall	0.99 ± 0.02	0.82 ± 0.05	0.99 ± 0.04	0.10 ± 0.10	0.70 ± 0.23	0.24 ± 0.10	0.41 ± 0.13	**1.00 ± 0.00**
	F1	0.99 ± 0.03	0.66 ± 0.05	0.98 ± 0.05	NA	0.65 ± 0.24	0.21 ± 0.08	0.43 ± 0.11	**0.99 ± 0.02**
	SHD	0.30 ± 0.54	8.58 ± 1.34	0.36 ± 1.05	12.84 ± 1.20	8.28 ± 5.73	14.28 ± 1.74	11.74 ± 2.67	**0.20 ± 0.45**

Data size	Metrics	Methods with **Three Confounders** on $\{x_1, x_2\}$, $\{x_5, x_6\}$, $\{x_8, x_9\}$, repectively (dim=10, mean ± standard deviation)							
		Direct	DirectKernel	ICA	RCD	CAMUV	PC	Parce	MMI (ours)
$n = 100$	Precision	0.61 ± 0.17	0.58 ± 0.06	0.71 ± 0.19	NA	0.41 ± 0.22	0.28 ± 0.09	0.44 ± 0.14	**0.88 ± 0.19**
	Recall	0.57 ± 0.23	0.78 ± 0.07	0.72 ± 0.19	0.00 ± 0.02	0.24 ± 0.15	0.30 ± 0.11	0.33 ± 0.13	**0.90 ± 0.18**
	F1	0.58 ± 0.19	0.67 ± 0.06	0.72 ± 0.19	NA	NA	0.29 ± 0.10	0.37 ± 0.13	**0.89 ± 0.18**
	SHD	8.04 ± 3.71	7.96 ± 1.70	5.72 ± 3.98	12.96 ± 0.20	12.18 ± 2.43	12.28 ± 2.20	11.50 ± 2.22	**2.32 ± 3.89**
$n = 200$	Precision	0.57 ± 0.08	0.54 ± 0.05	0.67 ± 0.22	NA	0.51 ± 0.13	0.23 ± 0.06	0.42 ± 0.12	**0.94 ± 0.11**
	Recall	0.68 ± 0.27	0.81 ± 0.10	0.71 ± 0.23	0.01 ± 0.02	0.43 ± 0.16	0.26 ± 0.08	0.37 ± 0.14	**0.97 ± 0.06**
	F1	0.69 ± 0.24	0.67 ± 0.09	0.69 ± 0.22	NA	0.63 ± 0.22	0.25 ± 0.07	0.39 ± 0.13	**0.95 ± 0.09**
	SHD	6.14 ± 4.63	8.14 ± 2.50	6.60 ± 4.73	13.10 ± 0.78	10.48 ± 2.59	13.36 ± 1.38	11.76 ± 2.35	**1.04 ± 2.04**
$n = 500$	Precision	0.92 ± 0.11	0.62 ± 0.05	**0.96 ± 0.11**	NA	0.58 ± 0.17	0.21 ± 0.07	0.45 ± 0.12	0.95 ± 0.07
	Recall	0.95 ± 0.10	0.86 ± 0.03	0.98 ± 0.09	0.08 ± 0.10	0.66 ± 0.20	0.26 ± 0.10	0.44 ± 0.15	**0.98 ± 0.03**
	F1	0.93 ± 0.10	0.72 ± 0.04	0.97 ± 0.10	NA	0.61 ± 0.17	0.23 ± 0.08	0.44 ± 0.12	**0.97 ± 0.05**
	SHD	1.48 ± 2.41	7.00 ± 1.10	0.76 ± 2.24	12.80 ± 1.28	9.06 ± 3.86	14.17 ± 1.54	11.48 ± 2.35	**0.66 ± 0.93**

B EXTRA SIMULATION

B.1 SHD.

In this section, we give the simulation result when $p = 10$. For the case without confounder, we first generate $p = 10$ random variables $\{x_0, \dots, x_9\}$ with size n as follows: $x_0 = e_0$, $x_1 = x_0 + e_1$, $x_2 = x_1 + e_1$, $x_3 = x_2 + e_3$, $x_4 = x_3 + e_4$, $x_5 = x_4 + x_2 + e_5$, $x_6 = x_5 + x_3 + x_1 + e_6$, $x_7 = x_6 + x_4 + e_7$, $x_8 = x_7 + e_8$, $x_9 = x_8 + e_9$ ($p = 10$) with $e_i \sim \text{Laplace}(0, 1/\sqrt{2})$. We show the results in the upper Table in the appendix. For the case with confounder, we generate the confounder as $f \sim 2U(0, 1/\sqrt{2}) - 1$ on $\{x_1, x_2\}$, the results are in the middle Table in the appendix. In addition, we also consider about multiple confounders $f_i \sim 2U(0, 1/\sqrt{2}) - 1$ for f_1 on $\{x_1, x_2\}$, f_2 on $\{x_5, x_6\}$, f_3 on $\{x_8, x_9\}$ as the lower Table in appendix. Whether a confounder is present, our LiNGAM-MMI achieves the best performance among the eight, especially when n is small.

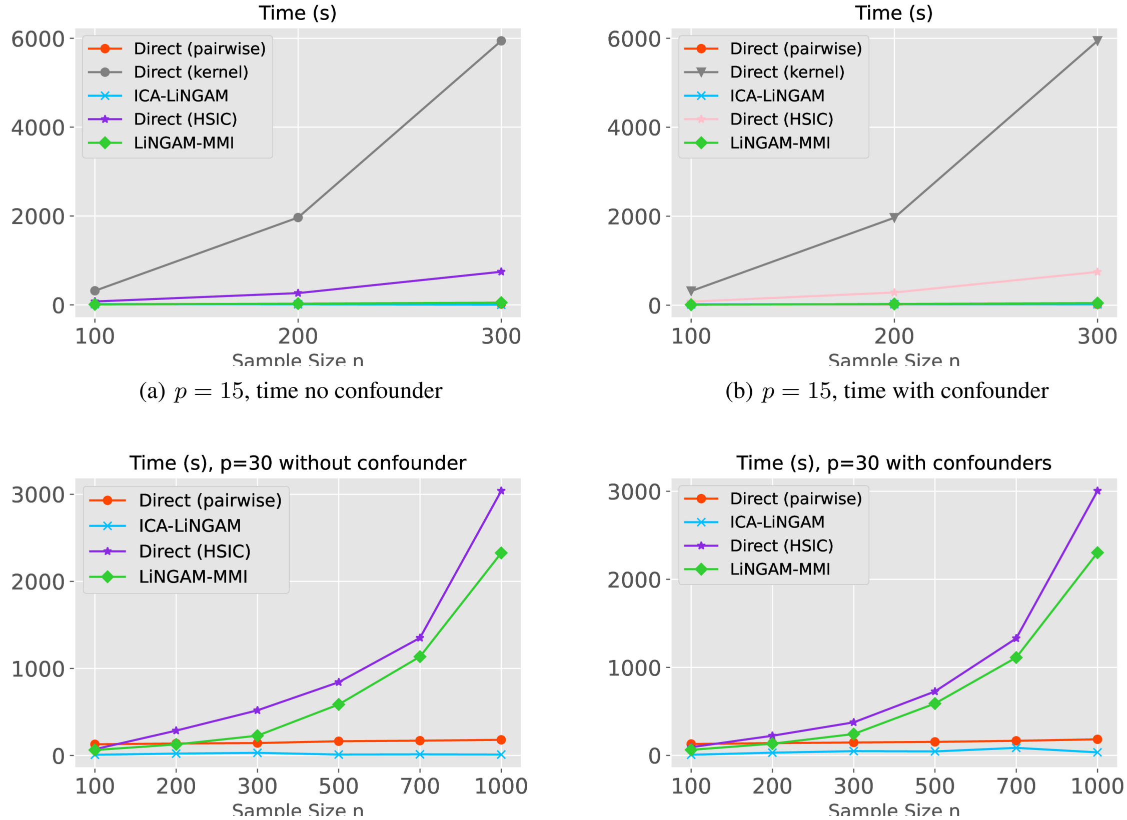


(a) $p = 15$, no confounder (A) (b) $p = 15$, no confounder (B) (c) $p = 15$, with confounder (A) (d) $p = 15$, with confounder (B)

Figure 1: Error ratio of Criterion A and B with the change of number n (the lower, the better) with copula entropy.

B.2 RUNNING TIME

In this section, we give the comparison running time among several methods and our LiNGAM-MMI when $p = 15, 30$ with 50 trials as in Figure 2.



(a) $p = 15$, time no confounder

(b) $p = 15$, time with confounder

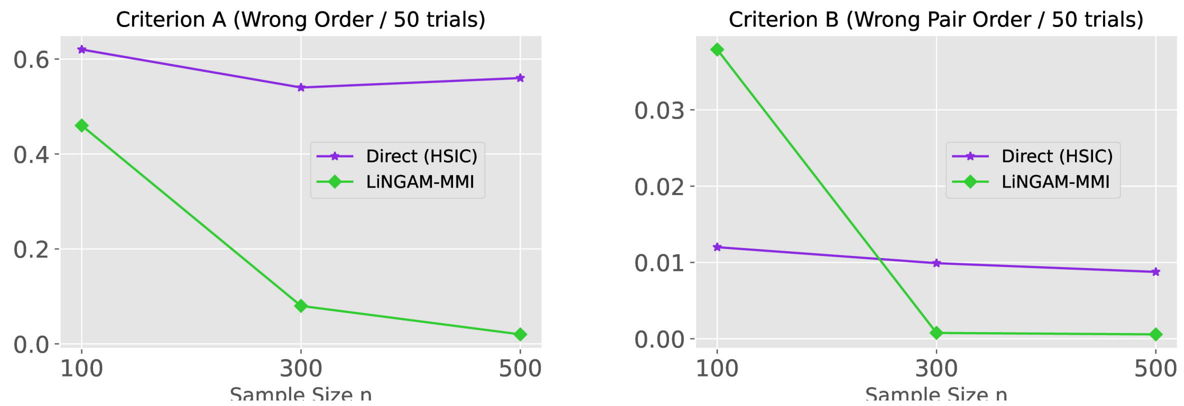
(c) $p = 30$, time no confounder

(d) $p = 30$, time with confounder

Figure 2: **With Confounder** Time (s) with change of number n (the lower the better).

B.3 LATENT CONFOUNDERS WITH NORMAL DISTRIBUTION

Give more examples of different distributions of latent confounders (arbitrary distribution is ok). We do the simulation when latent confounder is Normal distribution ($f \sim N(0, 1)$) as $p = 15$ in Figure 3. The true causal order is from $x_0 \to x_1, \ldots, \to x_{p-1} \to x_p$ with $x_i = x_{i-1} + e_i, i = 1, \ldots, p$. We set the confounder as f_1 on $\{x_2, x_3\}$, f_2 on $\{x_5, x_6\}$, f_3 on $\{x_8, x_9\}$, f_4 on $\{x_{12}, x_{13}\}$, respectively. Here we compare with DirectLiNGAM (HSIC).



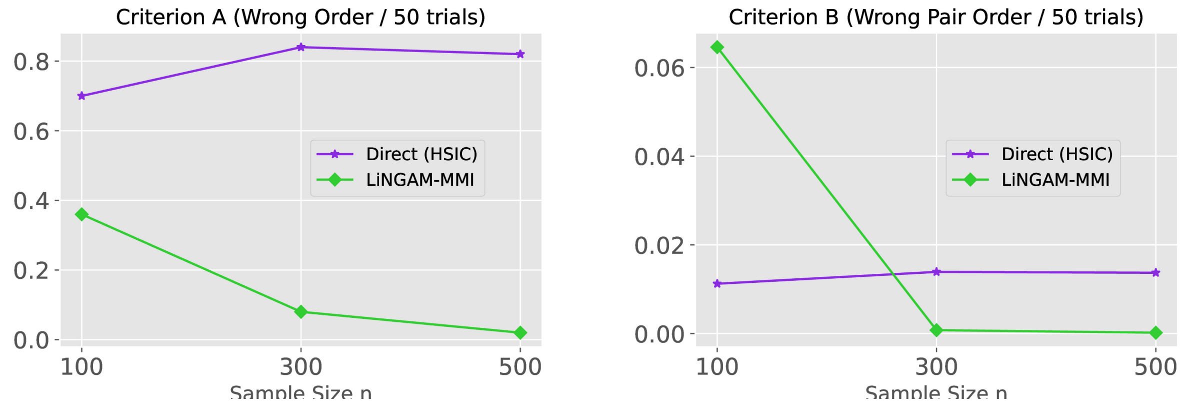
(a) Criterion A (Normal Confounders).

(b) Criterion B (Normal Confounders).

Figure 3: Error ratio between DirectLiNGAM with HSIC and LiNGAM-MMI.

B.4 LATENT CONFOUNDERS WITH EXPONENTIAL DISTRIBUTION

In this section, we do the simulation when the latent confounder is Exponential distribution $(f \sim Exp(1))$ as $p = 15$ in Figure 4. The true causal order is from $x_0 \rightarrow x_1, \ldots, \rightarrow x_{p-1} \rightarrow x_p$ with $x_i = x_{i-1} + e_i, i = 1, \ldots, p$ when confounder is not present. We set the confounder as f_1 on $\{x_2, x_3\}$, f_2 on $\{x_5, x_6\}$, f_3 on $\{x_8, x_9\}$, f_4 on $\{x_{12}, x_{13}\}$, respectively.



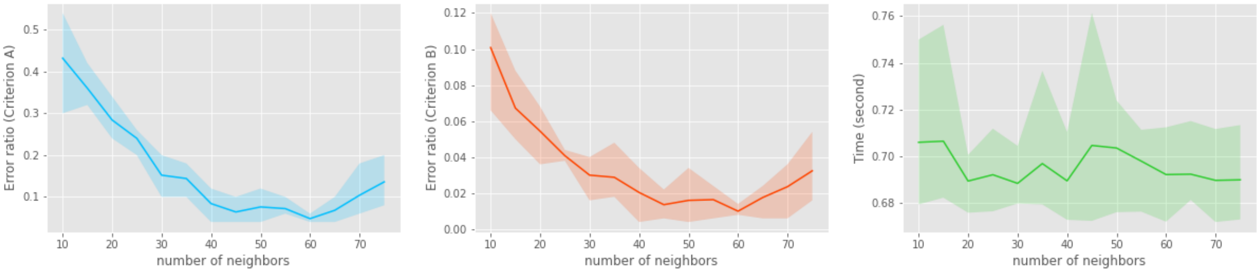
(a) Criterion A (Exponential Confounders).

(b) Criterion B (Exponential Confounders).

Figure 4: Error ratio between DirectLiNGAM with HSIC and LiNGAM-MMI.

B.5 SELECT HYPERPARAMETER k (NUMBER OF NEIGHBORS)

This section we choose the optimal hyperparameter k when $p = 5, n = 100$ and with 2 confounders as in Figure 5. From our experience, the hyperparameter k can be small ($k = 10$) with enough samples. When the sample size is small, we may choose the k in $[40, 60]$.



(a) Criterion A.

(b) Criterion B.

(c) Time (s).

Figure 5: The left is the Criterion A error ratio with the neighbor numbers, the middle is the Criterion B error ratio with the neighbor numbers (lower is better), and the right is the time with neighbor numbers.

References

Jian Ma and Zengqi Sun. Mutual information is copula entropy. *Tsinghua Science & Technology*, 16(1):51–54, 2011. ISSN 1007-0214. doi: https://doi.org/10.1016/S1007-0214(11)70008-6. URL `https://www.sciencedirect.com/science/article/pii/S1007021411700086`.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research*, 12(33):1225–1248, 2011. URL `http://jmlr.org/papers/v12/shimizu11a.html`.