Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Testing Directed Acyclic Graph via Structural, Supervised and Generative Adversarial Learning

Chengchun Shi, Yunzhe Zhou & Lexin Li

To cite this article: Chengchun Shi, Yunzhe Zhou & Lexin Li (2023): Testing Directed Acyclic Graph via Structural, Supervised and Generative Adversarial Learning, Journal of the American Statistical Association, DOI: 10.1080/01621459.2023.2220169

To link to this article: https://doi.org/10.1080/01621459.2023.2220169

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2023, VOL. 00, NO. 0, 1–14: Theory and Methods
https://doi.org/10.1080/01621459.2023.2220169

Testing Directed Acyclic Graph via Structural, Supervised and Generative Adversarial Learning

Chengchun Shi[a], Yunzhe Zhou[b], and Lexin Li[b]

[a]London School of Economics and Political Science, London, UK; [b]University of California at Berkeley, Berkeley, CA

ABSTRACT

In this article, we propose a new hypothesis testing method for directed acyclic graph (DAG). While there is a rich class of DAG estimation methods, there is a relative paucity of DAG inference solutions. Moreover, the existing methods often impose some specific model structures such as linear models or additive models, and assume independent data observations. Our proposed test instead allows the associations among the random variables to be nonlinear and the data to be time-dependent. We build the test based on some highly flexible neural networks learners. We establish the asymptotic guarantees of the test, while allowing either the number of subjects or the number of time points for each subject to diverge to infinity. We demonstrate the efficacy of the test through simulations and a brain connectivity network analysis. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received May 2021
Accepted May 2023

KEYWORDS
Brain connectivity networks; Directed acyclic graph; Generative adversarial networks; Hypothesis testing; Multilayer perceptron neural networks

1. Introduction

Directed acyclic graph (DAG) is an important tool to characterize pairwise associations among multivariate and high-dimensional random variables. It has been frequently used in a wide range of scientific applications. One example is gene regulatory network analysis in genetics (Sachs et al. 2005), where the time-course expression data of multiple genes are measured over multiple cellular samples through microarray or RNA sequencing, and the goal is to understand the regulatory activation or repression relations among different genes. Another example is brain effective connectivity analysis in neuroscience (Garg, Cecchi, and Rao 2011), where the time-course neural activities are measured at multiple brain regions for multiple experimental subjects through functional magnetic resonance imaging, and the goal is to infer the influences of brain regions exerting over each other under the stimulus.

There is a large body of literature studying penalized estimation of DAG given the observational data (see, e.g., Spirtes, Glymour, and Scheines 2000; van de Geer and Bühlmann 2013; Zheng et al. 2018; Yuan et al. 2019, among many others). These works all impose some specific model structures, most often, linear models or additive models. There have recently emerged a number of proposals in the computer science literature that used neural networks or reinforcement learning to tackle nonlinear models and to estimate the associated DAG (Yu et al. 2019; Zheng et al. 2020; Zhu, Ng, and Chen 2020). While all these works have made crucial contributions, DAG model *estimation* is an utterly different problem from DAG inference. By inference, we mean hypothesis testing of individual edges throughout this article. The two problems are closely related, and both can, in effect, identify important links of a DAG. Besides, DAG inference usually relies on DAG estimation as a precedent step. Nevertheless, estimation does not produce an explicit quantification of statistical significance as inference does. Bayesian networks have been proposed for DAG estimation and inference. However, computationally, it is extremely difficult to search through all possible graph structures in a Bayesian network (Chickering, Heckerman, and Meek 2004), and as a result, the dimension of the Bayesian network is often small (Friston 2011). There are very few frequentist inference solutions for inferring DAG structures. Only recently, Janková and van de Geer (2019) proposed a de-biased estimator to construct confidence intervals for the edge weights in a DAG, whereas Li, Shen, and Pan (2020) developed a constrained likelihood ratio test to infer individual edges or some given directed paths of a DAG. These works are probably the most relevant to our proposal. However, both have focused on Gaussian linear DAG, and cannot be easily extended to more general nonlinear DAG models. Moreover, all the above works considered the setting where the data observations are iid. Learning DAG from time-dependent data remains largely unexplored.

There is another body of literature studying conditional independence testing (CIT); see Li and Fan (2019), Shah and Peters (2020), Shi et al. (2021) and the references therein. CIT is closely related to DAG inference, and is to serve as a building block of our proposed testing procedure. On the other hand, naively performing CIT on two variables given the rest would fail to infer the directed edges of a DAG; see Section 2.2 for details. Besides, most CIT methods assume the data observations are independent, and are not suitable for the setting where the measurements are time-dependent.

CONTACT Lexin Li *lexinli@berkeley.edu* University of California at Berkeley, Berkeley, CA.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JASA*.

In this article, we propose a novel statistical testing procedure for the inference of individual links or some given paths in a large and general DAG. The new test hinges upon some highly flexible neural networks-based machine learning techniques. The associations among the random variables can be either linear or nonlinear, the variables themselves can be either continuous or discrete-valued, and the observed data can be time-dependent.

Methodologically, we employ a number of state-of-the-art deep learning techniques that are highly flexible and can capture nonlinear associations among high-dimensional variables. We begin with a new characterization of directed edges under the additive noise structure (Peters et al. 2014); see Theorem 1. Based on this characterization, we propose a new testing procedure that integrates three key deep learning ingredients: (a) a DAG structural learning method based on neural networks or reinforcement learning to estimate the DAG; (b) a supervised learning method based on neural networks to estimate the conditional mean; and (c) a distribution generator produced by generative adversarial networks (Goodfellow et al. 2014, GANs) to approximate the conditional distribution of the variables in the DAG. We further couple these deep learning tools with some hypothesis testing strategies, including data splitting and cross-fitting to ensure a valid size control, and constructing a doubly robust test statistic as the maximum of multiple transformation functions to improve the power.

Theoretically, we establish the asymptotic size and power guarantees for the proposed test. The data-splitting and cross-fitting strategy ensures that our test achieves a valid Type-I error control asymptotically under minimal conditions on those learning methods. As a result, our test procedure can work with a wide range of nonparametric estimators. Next, our DAG testing procedure requires a DAG estimation solution as a precedent step, which is common for almost all graph inference approaches (Cai 2017). However, we do not assume the ordering of the nodes is known a priori, but instead estimate this DAG ordering from the data using some DAG structural learning method. To establish the consistency of the proposed test, we require this ordering is consistently estimated; see condition (C1). Nevertheless, this order consistency is much weaker than requiring the initial DAG estimator to be selection consistent, or to satisfy the sure screening property. In other words, we only require a reasonably good initial estimator of DAG, which is order consistent but not necessarily selection consistent. We then develop a testing procedure that produces an explicit quantification of statistical significance for each individual link, and we show the test has the desired size and power guarantees. We also prove that the estimator from the DAG structural learning method we employ is indeed order consistent. Meanwhile, we discuss the impact on our test when this order consistency condition is not satisfied. Finally, for our theoretical analysis, we introduce a bidirectional asymptotic framework that allows either the number of subjects, or the number of time points for each subject, to diverge to infinity. This is useful for different types of applications. There are plenty of studies where the interest is about the general population, and thus it is reasonable to let the number of subjects or samples to diverge. Meanwhile, there are plenty of other applications, for example, neuroimaging-based brain networks studies, where the number of subjects is almost always limited, but the scanning time and the temporal resolution can greatly increase. For those applications, it is more suitable to let the number of time points to diverge.

Our proposal is innovative and makes useful contributions in several ways.

First, rigorous inference of directed edges in DAG is a vital but also a long-standing open question. The existing solutions rely on particular model structures such as linear or additive models, and mostly deal with iid data. Such requirements can be restrictive in numerous applications, since the actual relations may be nonlinear and the data are correlated. By contrast, we only require an additive noise structure. To the best of our knowledge, our work is the first frequentist hypothesis testing solution for a general DAG with time-dependent data.

Second, we employ modern deep learning techniques such as neural networks and GANs to help address a classical statistical hypothesis testing problem. Such modern learning methods serve as nonparametric learners, and conceptually, play a similar role as splines and reproducing kernels. Meanwhile, they are often more flexible and can handle more complex data structures. With increasingly efficient implementations of these methods and improved understandings of their theoretical properties (e.g., Bauer and Kohler 2019; Farrell, Liang, and Misra 2021), this family of deep learning methods offer a powerful set of tools for classical statistical problems. Our proposal can be viewed as one of the early examples of harnessing such power, as the use of these deep learning techniques allows us to accurately estimate the DAG structure, the conditional means, as well as the distribution functions, and to improve the power of the test.

Third, even though the individual learning components such as neural networks, GANs and cross-fitting are not completely new, how to integrate them properly and effectively into a test with desired theoretical guarantees is highly nontrivial, and is one of the main contributions of this article. In effect, our proposed test achieves a parametric convergence rate and a parametric power guarantee while using nonparametric estimators. This is made possible mainly due to the innovative way we put together these learning components, which leads to a doubly robust test statistic (Tsiatis 2007), in the sense that the proposed statistic is consistent, as long as either the conditional mean function in (b), or the distribution generator in (c) is correctly specified. In our solution, we propose to estimate both the conditional mean and the distribution generator fully nonparametrically. As such, the convergence rate of the two estimators, denoted by κ_1 and κ_2, respectively, may each be slower than the parametric rate. Nevertheless, we only require $\kappa_1 + \kappa_2 > 1/2$, which is totally achievable for the multilayer perceptron models and GANs; see the discussion after condition (C4). The key idea of our theoretically analysis is to show the bias of the estimating equation grows faster than the parametric rate. Thanks to the double robustness property of the test statistic, if we replace either estimator with its oracle value, the bias would be equal to zero. This observation, together with the Neyman orthogonality property of the estimating equation, ensures that the bias can be represented as a product of the difference between the two nonparametric estimators and their oracle values. Consequently, when $\kappa_1 + \kappa_2 > 1/2$, the test statistic converges at a parametric rate, the corresponding test controls the Type-I error, and has a parametric power guarantee. We comment that, in their seminal work on double/debiased machine learning, Chernozhukov

et al. (2018) proposed to combine two machine learning estimators to infer the average treatment effect, which they showed to achieve a parametric convergence rate, even though each of the machine learning estimator converges at a nonparametric rate. Our result is similar in spirit as theirs, but targets a completely different problem, and thus is the first of its kind for DAG inference.

The rest of the article is organized as follows. We formally define the hypotheses, along with the model and data structure, in Section 2. We develop the testing procedure in Section 3, and establish the theoretical properties in Section 4. We study the empirical performance of the test through simulations and a real data example in Sections 5 and 6. We relegate several extensions, additional results, and all technical proofs to the Supplementary Appendix.

2. Problem Formulation

In this section, we first present the DAG model, based on which we formally define our hypotheses. We next propose an equivalent characterization of the hypotheses, for which we develop our testing procedure. Finally, we detail the data structure.

2.1. DAG Model

Consider d random variables $X = (X_1, \ldots, X_d)^\top$, each with a finite fourth moment. We use a directed graph to characterize the relationships among these variables, where a node of the graph corresponds to a variable in X. For two nodes $i, j \in \{1, \ldots, d\}$, if an arrow is drawn from i to j, that is, $i \to j$, then X_i is called a parent of X_j, and X_j a child of X_i. A directed path in the graph is a sequence of distinct nodes $i_1, \ldots, i_{d'}$, such that there is a directed edge $i_k \to i_{k+1}$ for all $k = 1, \ldots, d' - 1$. If there exists a directed path from i to j, then X_i is called an ancestor of X_j, and X_j a descendant of X_i. For node X_j, let PA_j, DS_j and AC_j denote the set of indices of the parents, descendants, and ancestors of X_j, respectively. Moreover, let $X_{\mathcal{M}}$ denote the sub-vector of X formed by those whose indices are in a subset $\mathcal{M} \subseteq \{1, \ldots, d\}$.

To rigorously formulate our problem, we make two assumptions.

(A1) The directed graph is acyclic; that is, no variable is an ancestor of itself.
(A2) The DAG is identifiable from the joint distribution of X.

Condition (A1) has been commonly imposed in directed graph analysis. It does not permit any variable to be its own ancestor. As a result, the relationship between any two variables is unidirectional. Condition (A2) helps simplify the problem, and avoids dealing with the equivalence class of DAG. This condition is again frequently imposed in the DAG estimation literature (Zheng et al. 2018; Yuan et al. 2019; Li, Shen, and Pan 2020; Zheng et al. 2020). We discuss the extension to the equivalence class in Section S1.4 of the Appendix.

We consider a class of structural equation models that follow an additive noise structure,

$$X_j = f_j(X_{\mathrm{PA}_j}) + \varepsilon_j, \qquad \text{for any } j = 1, \ldots, d, \tag{1}$$

where $\{f_j\}_{j=1}^d$ are a set of continuous functions, and $\{\varepsilon_j\}_{j=1}^d$ are a set of independent zero mean random errors. Model (1) permits a fairly flexible structure. For instance, if each f_j is a linear function, then (1) reduces to a linear structural equation model. If each f_j is an additive function, that is, $f_j(X_{\mathrm{PA}_j}) = \sum_{k \in \mathrm{PA}_j} f_{j,k}(X_k)$, then (1) becomes an additive model. In our test, we do *not* impose linear or additive model structures. Moreover, we can easily extend the proposed test to the setting of generalized linear model, where the X_j can be either continuous or discrete-valued. We discuss such an extension in Section S1.3 of the Appendix.

Under model (1), the corresponding DAG is identifiable under some reasonable conditions. We consider three examples to discuss explicitly those conditions.

Example 1 (Gaussian graphical model). Suppose $X_1, \ldots, X_d$ are jointly normal, and model (1) becomes $X_j = W_j^\top X_{\mathrm{PA}_j} + b_j + \varepsilon_j$, for some W_j and b_j. Then the corresponding DAG is identifiable, if the variance of the random error ε_j is the same for all $j = 1, \ldots, d$ (Bühlmann, Peters, and Ernest 2014, Theorem 1).

Example 2 (Nonlinear graphical model with Gaussian noise). Suppose $\varepsilon_1, \ldots, \varepsilon_d$ are jointly normal, but $X_1, \ldots, X_d$ are not. Then the corresponding DAG is identifiable, if each f_j is three times differentiable and not linear in any of its arguments (Peters et al. 2014, Corollary 31).

Example 3 (Nonlinear graphical model with general noise). Suppose neither X_j nor ε_j is normal. Then the corresponding DAG is identifiable, if each f_j is non-constant in each of its arguments, and (1) is a restricted additive noise model (Peters et al. 2014, Definition 27).

2.2. Hypotheses and Equivalent Characterization

We next formally define the hypotheses we target, then give an equivalent characterization. For a given pair of nodes $(j, k), j, k = 1, \ldots, d, j \neq k$, we aim at the hypotheses:

$$H_0(j,k) : k \notin \mathrm{PA}_j, \quad \text{versus} \quad H_1(j,k) : k \in \mathrm{PA}_j. \tag{2}$$

When the alternative hypothesis holds, there is a link from X_k to X_j. In the following, we mainly focus on testing an individual link $H_0(j,k)$. We discuss the extension of testing a directed pathway, or a union of links, in Sections S1.1 and S1.2 of the Appendix.

We next consider a pair of hypotheses that involve two variables that are *conditionally independent* (CI). The new hypotheses are closely related to (2), but are *not* exactly the same.

$$\begin{aligned} &H_0^*(j,k) : X_k \text{ and } X_j \text{ are CI given the rest of variables}, \quad \text{versus} \\ &H_1^*(j,k) : X_k \text{ and } X_j \text{ are } \textit{not} \text{ CI given the rest of variables}. \end{aligned} \tag{3}$$

We point out that, testing for (3) is generally *not* the same as testing for (2). To elaborate this, we consider a three-variable DAG with a v-structure.

Example 4 (v-structure). Consider three random variables X_1, X_2, X_3 that form a v-structure, as illustrated in Figure 1(a), where X_1 and X_2 are the common parents of X_3. Even if X_1 and X_2 are *marginally* independent, they can be *conditionally* dependent given X_3. To better understand this, consider the following toy illustration. Either the ballgame or the rain could cause traffic jam, but they are uncorrelated. However, seeing

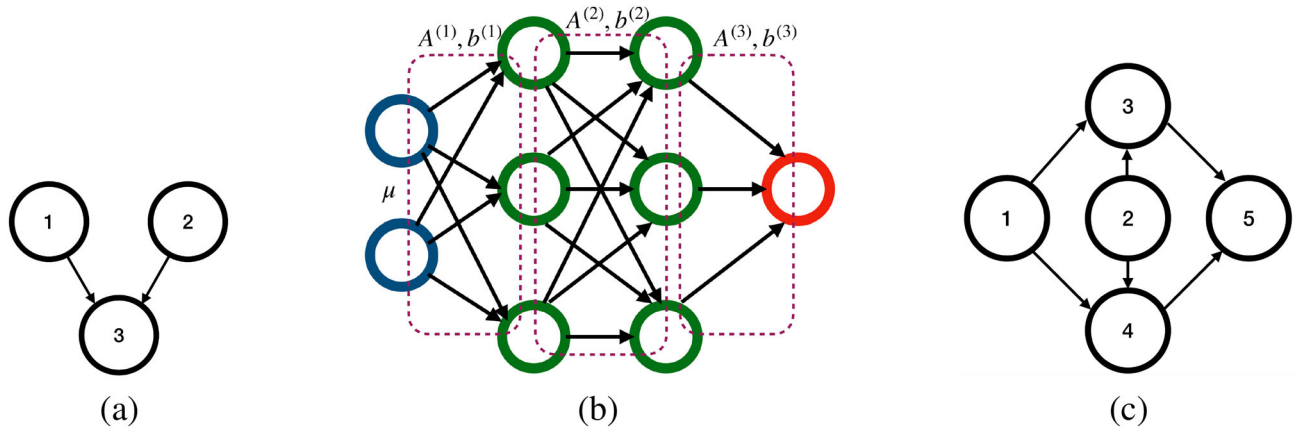

Figure 1. (a) A three-variable DAG with a v-structure; (b) A graphical illustration of a multilayer perceptron, with two hidden layers, $m_0 = 2, m_1 = m_2 = 3$, where u is the input, $A^{(\ell)}$ and $b^{(\ell)}$ denote the corresponding parameters to produce the linear transformation for the $(\ell - 1)$th layer; (c) A five-variable DAG.

traffic jam puts the ballgame and the rain in competition as a potential explanation. As such, these two events are conditionally dependent. Since X_2 is not a parent of X_1, both $H_0(1,2)$ and $H_1^*(1,2)$ hold. Consequently, testing for (3) can have an inflated Type-I error for testing (2).

In this example, we see the reason that testing for (3) is not the same as for (2) is because the conditioning set of X_1 and X_2 contains their common descendant X_3. This key observation motivates us to consider a variant of (3), which we show is equivalent to (2) under certain conditions. We also remark that missing links in a DAG correspond to specific conditional independence between variables, but are not equivalent to marginal independence in general.

Specifically, for a given set of indices $\mathcal{M} \subseteq \{1, \ldots, d\}$ such that $j \notin \mathcal{M}$, and letting $X_{\mathcal{M}-\{k\}}$ denote the set of variables in $\mathcal{M} - \{k\}$, we consider the hypotheses:

$$\begin{aligned} &H_0^*(j,k|\mathcal{M}) : X_k \text{ and } X_j \text{ are CI given } X_{\mathcal{M}-\{k\}}, \quad \text{versus} \\ &H_1^*(j,k|\mathcal{M}) : X_k \text{ and } X_j \text{ are } \textit{not} \text{ CI given } X_{\mathcal{M}-\{k\}}, \end{aligned} \tag{4}$$

Proposition 1. For a given pair of nodes (j,k) such that $j \in \mathrm{DS}_k$, $j,k = 1,\ldots,d$, and for any $\mathcal{M}$ such that $j \notin \mathcal{M}$, $\mathrm{PA}_j \subseteq \mathcal{M}$ and $\mathcal{M} \cap \mathrm{DS}_j = \emptyset$, testing (4) is equivalent to testing (2).

Proposition 1 forms the basis for our test. That is, to infer the directed links, we first restrict our attention to the pairs (j,k) such that $j \in \mathrm{DS}_k$. Apparently, $H_0(j,k)$ does not hold when $j \notin \mathrm{DS}_k$. Next, when devising a conditional independence test for $H_0(j,k)$, the conditioning set $\mathcal{M}$ is supposed to contain the parents of node j, but *cannot* contain any common descendants of j,k. Under these conditions, we establish the equivalence between (4) and (2). A similar idea of using CI tests for DAG structural learning was employed in Spirtes, Glymour, and Scheines (2000) too.

Next, we develop a test statistic for the hypotheses (4). We introduce a key quantity. Let h denote a square-integrable function that takes X_k and $X_{\mathcal{M}-\{k\}}$ as the input. Define

$$\begin{aligned} I(j,k|\mathcal{M};h) = \mathbb{E}\left\{X_j - \mathbb{E}\left(X_j|X_{\mathcal{M}-\{k\}}\right)\right\}\left[h\left(X_k, X_{\mathcal{M}-\{k\}}\right)\right. \\ \left. -\mathbb{E}\left\{h\left(X_k, X_{\mathcal{M}-\{k\}}\right)|X_{\mathcal{M}-\{k\}}\right\}\right]. \end{aligned}$$

Under the additive noise model (1), the next theorem connects this quantity with the null hypothesis $H_0^*(j,k|\mathcal{M})$ in (4). Together with Proposition 1, it shows that $I(j,k|\mathcal{M};h)$ can serve as a test statistic for (4), and equivalently, for (2) that we target.

Theorem 1. Suppose (1) holds. For a given pair of nodes (j,k) such that $j \in \mathrm{DS}_k$, $j,k = 1,\ldots,d$, for any $\mathcal{M}$ such that $j \notin \mathcal{M}$, $\mathrm{PA}_j \subseteq \mathcal{M}$ and $\mathcal{M} \cap \mathrm{DS}_j = \emptyset$, the null hypothesis $H_0^*(j,k|\mathcal{M})$ in (4) is equivalent to $\sup_h |I(j,k|\mathcal{M};h)| = 0$ where the supremum is taken over all square-integrable functions h.

Theorem 1 immediately suggests a possible testing procedure for (4). That is, we first employ a DAG estimator to learn the ancestors and descendants for node j. We then consider a natural choice for h, where $h\left(X_k, X_{\mathcal{M}-\{k\}}\right) = X_k$. Then $I(j,k|\mathcal{M};h)$ becomes

$$\begin{aligned} I(j,k|\mathcal{M};h) = \mathbb{E}\left\{X_j - \mathbb{E}\left(X_j|X_{\mathcal{M}-\{k\}}\right)\right\} \\ \left\{X_k - \mathbb{E}\left(X_k|X_{\mathcal{M}-\{k\}}\right)\right\}. \end{aligned} \tag{5}$$

By Theorem 1, under the null hypothesis $H_0^*(j,k|\mathcal{M})$, a consistent estimator for (5) should be close to zero. A Wald type test can then be devised with iid data. That is, we first obtain an estimator $\widehat{I}_{j,k}$ for $I(j,k|\mathcal{M};h)$, by plugging in the estimators of the conditional mean functions, $\widehat{\mathbb{E}}\left(X_j|X_{\mathcal{M}-\{k\}}\right)$ and $\widehat{\mathbb{E}}\left(X_k|X_{\mathcal{M}-\{k\}}\right)$. We then get an estimator of its asymptotic variance $\widehat{\sigma}_{j,k}^2$, and obtain the Wald type test statistic, $\sqrt{N}\widehat{\sigma}_{j,k}^{-1}\widehat{I}_{j,k}$, where N is the number of samples. Such a test is similar in spirit as the tests of Zhang, Zhou, and Guan (2018) and Shah and Peters (2020). Since it involves estimation of two conditional mean functions, we refer to it as the *double regression-based test*. We later numerically compare our proposed test with this test.

On the other hand, this double regression-based test has some limitations. One is that it requires the set $\mathcal{M}$ to be fixed. To meet the requirement in Proposition 1, $\mathcal{M}$ needs to be determined in a data-adaptive way. The resulting test may not control the type-I error due to the dependence between $\mathcal{M}$ and the estimator of the mean functions in $\widehat{I}_{j,k}$. Another limitation is that it may not have a sufficient power to detect $H_1(j,k)$. As an illustration, we revisit Example 4. For this example, consider the structural equation model: $X_1 = \varepsilon_1$, $X_2 = \varepsilon_2$, and $X_3 = X_1^2 + X_2 + \varepsilon_3$. Under this model, $H_1(1,3)$ holds. Meanwhile, $I(1,3) = \mathbb{E}(X_3 - X_2)X_1 = \mathbb{E}\varepsilon_1^3$. When the distribution of ε_1 is

symmetric, $I(1,3)=0$, despite the fact that X_1 is a parent of X_3. As such, for this example, the double regression-based test is to have no power at all.

To address the first limitation, we employ the sample splitting strategy to ensure its size control. To address the second limitation, we consider multiple transformation functions h, instead of a single h, to improve the power. We detail our idea in Section 3.

2.3. Time-Dependent Observational Data

Throughout this article, we use X to denote the population variables, and $\mathbb{X}$ to denote the data realizations. Suppose the data come from an observational study, and are of the form, $\{\mathbb{X}_{i,t,j} : i=1,\ldots,N, t=1,\ldots,T_i, j=1,\ldots,d\}$, where i indexes the ith subject, t indexes the tth time point, and j indexes the jth random variable. Suppose there are totally N subjects, with T_i observations for the ith subject. Write $\mathbb{X}_{i,t}=(\mathbb{X}_{i,t,1},\ldots,\mathbb{X}_{i,t,d})^\top$, $i=1,\ldots,N, t=1,\ldots,T_i$. We consider the following data structure.

(B1) Across subjects, the measurements $\mathbb{X}_{1,t},\ldots,\mathbb{X}_{N,t}$ are iid.
(B2) Across time points, the random vectors $\mathbb{X}_{i,1},\ldots,\mathbb{X}_{i,T_i}$ are stationary.
(B3) For any i,t, $\mathbb{X}_{i,t,1},\ldots,\mathbb{X}_{i,t,d}$ are DAG-structured. In addition, their joint distribution is the same as that of $X_1,\ldots,X_d$.

Condition (B1) is reasonable, as the subjects are usually independent from each other. We do not study the scenario where the data come from the same families or clusters. Condition (B2) about the stationarity is common in numerous applications such as brain connectivity analysis (Bullmore and Sporns 2009; Qiu et al. 2016; Wang et al. 2016). Condition (B3) brings the data into the DAG framework that we study. Note that (B3) does not allow directed edges from past to future observations. Meanwhile, we discuss the extensions of our test for nonstationary DAG, or for past to future edges, in Section S1.5 of the Appendix.

3. Testing Procedure

In this section, we develop an inferential procedure for the hypotheses in (2) for a given pair (j,k), through (4), given the observational data $\mathbb{X}_{i,t}$. We first present the main ideas and the complete procedure, then detail the major steps. As our test is based on Structural learning, sUpervised learning, and Generative AdveRsarial networks, we call our method SUGAR.

3.1. The Main Algorithm

Our main idea is to construct a series of measures $\{I(j,k|\mathcal{M};h_b) : b=1,\ldots,B\}$, for a large number of transformation functions $h_1,\ldots,h_B$, then take the maximum of some standardized version of $I(j,k|\mathcal{M};h_b)$. Toward that goal, our test involves three key components:

(a) A DAG structural learning method to learn the set of indices $\mathcal{M}$ that satisfy Proposition 1;
(b) A supervised learning method to estimate the conditional mean function $\mathbb{E}\left(X_j|X_{\mathcal{M}-\{k\}}\right)$;
(c) A distribution generator to approximate the conditional distribution of the variables.

For (a), we apply a structural learning algorithm to learn the underlying DAG $\mathcal{G}$ corresponding to X. The input of this step is the observed data $\{\mathbb{X}_{i,t,j} : i=1,\ldots,N, t=1,\ldots,T_i, j=1,\ldots,d\}$, and the output is the estimated DAG. We then set $\mathcal{M}$ as the estimated set of ancestors of X_j. To capture possible sparsity and nonlinear associations in $\mathcal{G}$, we employ the DAG estimation method of Zheng et al. (2020). See Section 3.3 for details.

For (b), we employ a supervised learning algorithm. The input of this step is $X_{\mathcal{M}-\{k\}}$ that serves as the "predictors," and X_j that serves as the "response," and the output is the estimated mean function $\widehat{\mathbb{E}}\left(X_j|X_{\mathcal{M}-\{k\}}\right)$. We employ a multilayer perceptron learner, which has a good capacity of estimating complex high-dimensional mean, and the estimator has the desired consistency guarantees (Farrell, Liang, and Misra 2021). See Section 3.4 for details.

For (c), we propose to use generative adversarial networks (Goodfellow et al. 2014, GANs) to approximate the conditional distribution of X_k given $X_{\mathcal{M}-\{k\}}$. The input of this step is $\mathbb{X}_{i,t,\mathcal{M}-\{k\}}$ and multivariate Gaussian noise vectors, and the output is the learnt generator model, with a set of M pseudo samples $\widetilde{\mathbb{X}}_{i,t,k}^{(s,m)}$, $m=1,\ldots,M$, that have a similar distribution as the training samples. We employ a generator model with the Sinkhorn divergence loss (Genevay, Peyré, and Cuturi 2018) to mitigate the potential bias of GANs. See Section 3.5 for details.

Given the generated pseudo samples, we then proceed to estimate the conditional mean function $\mathbb{E}\left\{h_b\left(X_k,X_{\mathcal{M}-\{k\}}\right)|X_{\mathcal{M}-\{k\}}\right\}$ in (5), and construct the corresponding test statistic. We also incorporate the data-splitting and cross-fitting strategy (Romano and DiCiccio 2019), to ensure a valid Type-I error control for the test under minimal conditions for the above three learners. Specifically, we randomly split the samples into two equal halves $\mathcal{I}_1\cup\mathcal{I}_2$, where $\mathcal{I}_s$ denotes the set of subsample indices, $s=1,2$. We then compute the three learners in (a) to (c) using each half of the data separately. Based on these learners, we next use cross-fitting to estimate $\{I(j,k|\mathcal{M};h_b)\}_{b=1}^B$, and their associated standard deviations. We construct our test statistic as the largest standardized version of $I(j,k|\mathcal{M};h_b)$ in the absolute value. This leads to two Wald-type test statistics, one for each half of the data. Finally, we derive the p-values based on Gaussian approximation, and reject the null when either one of the p-value is smaller than $\alpha/2$. By Bonferroni's inequality, this yields a valid α-level test. See Section 3.2 for details.

A summary of the proposed testing procedure is given in Algorithm 1.

3.2. Test Statistic and p-value

We begin with the presentation of our test, including the test statistic and the computation of the p-value, which are built on the three learners in (a) to (c) that we discuss in detail later.

First, for each half of the data, $s=1,2$, we begin with a bounded function class $\mathbb{H}^{(s)}=\left\{h_\omega^{(s)} : \omega\in\Omega^{(s)}\right\}$, indexed by some parameter ω. In our implementation, we consider the class of characteristic functions of X_k,

$$\mathbb{H}^{(1)}=\mathbb{H}^{(2)}=\mathbb{H}=\left\{\cos(\omega X_k),\ \sin(\omega X_k) : \omega\in\mathbb{R}\right\}. \quad (6)$$

Algorithm 1 Testing procedure for a given edge (j,k).

Step 1. Randomly split the data into two equal halves, $\{\mathbb{X}_{i,t,k}\}_{i\in\mathcal{I}_s,t=1,\ldots,T_i}$, $s=1,2$.

Step 2. For each half of the data, $s=1,2$,

(2a) Apply the structural learning method (9) to estimate the DAG $\mathcal{G}$. Denote the estimated set of ancestors of X_j by $\widehat{\mathrm{AC}}_j^{(s)}$. Set $\mathcal{M}^{(s)}=\widehat{\mathrm{AC}}_j^{(s)}-\{k\}$.

(2b) If $k\notin\widehat{\mathrm{AC}}_j^{(s)}$, return the p-value, $p^{(s)}(j,k)=1$.

Step 3. For $s=1,2$, apply the supervised learning method (10) to estimate the conditional mean function $\mathbb{E}\big(X_j|X_{\mathcal{M}^{(s)}}\big)$, and denote the estimator by $\widehat{g}^{(s)}$.

Step 4. For $s=1,2$, apply the GANs method to learn a generator model to approximate the conditional distribution of X_k given $X_{\mathcal{M}^{(s)}-\{k\}}$. It returns the learnt generator $\mathbb{G}^{(s)}$, and a set of pseudo samples $\big\{\widetilde{\mathbb{X}}_{i,t,k}^{(s,m)}\big\}_{i\in\mathcal{I}_s,t=1,\ldots,T_i,m=1,\ldots,M}$.

Step 5. Construct the test statistic:

(5a) Randomly generate B functions $\big\{h_b^{(s)}\big\}_{b=1}^B$ from the class $\mathbb{H}^{(s)}$ in (6).

(5b) For each (s,b), construct two standardized measures, $\widehat{T}_{b,\mathrm{CF}}^{(s)}$ and $\widehat{T}_{b,\mathrm{NCF}}^{(s)}$, with and without cross-fitting, using (7).

(5c) Select the index, $\widehat{b}^{(s)}=\arg\max_{b\in\{1,\ldots,B\}}\big|\widehat{T}_{b,\mathrm{NCF}}^{(s)}\big|$, based on the measure without cross-fitting.

(5d) Set the test statistic as $\widehat{T}_{\widehat{b}^{(s)},\mathrm{CF}}^{(s)}$, based on the measure with cross-fitting.

Step 6. Return the p-value:

(6a) Compute the p-value, $p^{(s)}(j,k)=2\mathbb{P}\big\{Z_0\geq\big|\widehat{T}_{\widehat{b}^{(s)},\mathrm{CF}}^{(s)}\big|\big\}$, for each half of the data, $s=1,2$, where Z_0 is a standard normal random variable.

(6b) Return $p(j,k)=2\min\big\{p^{(1)}(j,k),p^{(2)}(j,k)\big\}$.

We note that (6) is not able to approximate the entire class of square integrable functions. Nevertheless, our numerical experiments have found that setting $\mathbb{H}^{(s)}$ according to (6) results in a good power empirically. Moreover, we note that one may set $\mathbb{H}^{(s)}$ to the class of characteristic functions of $(X_k,X_{\mathcal{M}^{(s)}})$. By the Fourier Theorem (Siebert 1986), this alternative choice can approximate any square integrable function h, and the resulting test is consistent against all alternatives. We choose (6) for its simplicity as well as good empirical performance. Without loss of generality, we choose an even number for the total number of transformation functions B. We randomly generate iid standard normal variables $\omega_1,\ldots,\omega_{B/2}$, and set

$$h_b^{(s)}\left(X_k,X_{\mathcal{M}^{(s)}}\right)=\begin{cases}\cos(\omega_b X_k), & \text{for } b=1,\ldots,B/2,\\ \sin(\omega_b X_k), & \text{for } b=B/2+1,\ldots,B.\end{cases}$$

Next, for each pair of (s,b), $b=1,\ldots,B, s=1,2$, let $\widehat{\mathrm{AC}}_j^{(s)}$, $\mathcal{M}^{(s)}$, $\widehat{g}^{(s)}$, and $\{\widetilde{\mathbb{X}}_{i,t,k}^{(s,m)}\}$ denote the estimated set of ancestors of X_j, the estimated set of indices $\mathcal{M}$, the estimated conditional mean function, and the generated pseudo samples, obtain from the components (a)–(c), respectively. We compute two estimators $\widehat{I}_{b,\mathrm{CF}}^{(s)}$ and $\widehat{I}_{b,\mathrm{NCF}}^{(s)}$ for the measure $I\left(j,k|\widehat{\mathrm{AC}}_j^{(s)},h_b^{(s)}\right)$, one *with* cross-fitting, and the other *without* cross-fitting. Specifically, we compute

$$\widehat{I}_{b,\mathrm{CF}}^{(s)}=\Big(\textstyle\sum_{i\in\mathcal{I}_s^c}T_i\Big)^{-1}\Big(\sum_{i\in\mathcal{I}_s^c}I_{i,t,b}^{(s)}\Big),$$
$$\widehat{I}_{b,\mathrm{NCF}}^{(s)}=\Big(\textstyle\sum_{i\in\mathcal{I}_s}T_i\Big)^{-1}\Big(\sum_{i\in\mathcal{I}_s}I_{i,t,b}^{(s)}\Big),$$

where

$$I_{i,t,b}^{(s)}=\Big\{\mathbb{X}_{i,t,j}-\widehat{g}^{(s)}\left(\mathbb{X}_{i,t,\mathcal{M}^{(s)}}\right)\Big\}$$
$$\left\{h_b^{(s)}\left(\mathbb{X}_{i,t,k},\mathbb{X}_{i,t,\mathcal{M}^{(s)}}\right)-\frac{1}{M}\sum_{m=1}^{M}h_b^{(s)}\left(\widetilde{\mathbb{X}}_{i,t,k}^{(s,m)},\mathbb{X}_{i,t,\mathcal{M}^{(s)}}\right)\right\},$$

and M is the total number of pseudo samples. We note that, for $\widehat{I}_{b,\mathrm{NCF}}^{(s)}$, we use the same subset of data to learn the graph, the generator, the condition mean function, and to construct $I_{i,t,b}^{(s)}$. By contrast, for $\widehat{I}_{b,\mathrm{CF}}^{(s)}$, the data used for the DAG learner, the conditional mean learner and the generator are independent from the data used to construct $I_{i,t,b}^{(s)}$.

Next, we compute the corresponding standard errors $\widehat{\sigma}_{b,\mathrm{CF}}^{(s)}$ and $\widehat{\sigma}_{b,\mathrm{NCF}}^{(s)}$ for $\widehat{I}_{b,\mathrm{CF}}^{(s)}$ and $\widehat{I}_{b,\mathrm{NCF}}^{(s)}$, respectively. Since our data are time-dependent, the usual sample variance would not be a consistent estimator. Therefore, we employ the batched estimator common in time series analysis (Carlstein 1986). That is, we divide the data associated with each subject into non-overlapping batches, with each batch containing at most K observations. For simplicity, suppose T_i is divisible by K for all $i=1,\ldots,N$. We obtain the following standard error estimators,

$$\widehat{\sigma}_{b,\mathrm{CF}}^{(s)}=\left[\frac{K}{\sum_{i\in\mathcal{I}_s^c}T_i}\sum_{i\in\mathcal{I}_s^c}\sum_{k=1}^{T_i/K}\left\{\frac{\sum_{t=(k-1)K+1}^{kK}\left(I_{i,t,b}^{(s)}-\widehat{I}_{b,\mathrm{CF}}^{(s)}\right)}{\sqrt{K}}\right\}^2\right]^{1/2},$$

$$\widehat{\sigma}_{b,\mathrm{NCF}}^{(s)}=\left[\frac{K}{\sum_{i\in\mathcal{I}_s}T_i}\sum_{i\in\mathcal{I}_s}\sum_{k=1}^{T_i/K}\left\{\frac{\sum_{t=(k-1)K+1}^{kK}\left(I_{i,t,b}^{(s)}-\widehat{I}_{b,\mathrm{NCF}}^{(s)}\right)}{\sqrt{K}}\right\}^2\right]^{1/2}.$$

Putting $\widehat{I}_{b,\mathrm{CF}}^{(s)}$ and $\widehat{I}_{b,\mathrm{NCF}}^{(s)}$ together with their standard error estimators, we obtain two standardized measures,

$$\begin{aligned}\widehat{T}_{b,\mathrm{CF}}^{(s)}&=\sqrt{\textstyle\sum_{i\in\mathcal{I}_s^c}T_i}\left(\widehat{\sigma}_{b,\mathrm{CF}}^{(s)}\right)^{-1}\widehat{I}_{b,\mathrm{CF}}^{(s)},\quad\text{and}\\ \widehat{T}_{b,\mathrm{NCF}}^{(s)}&=\sqrt{\textstyle\sum_{i\in\mathcal{I}_s}T_i}\left(\widehat{\sigma}_{b,\mathrm{NCF}}^{(s)}\right)^{-1}\widehat{I}_{b,\mathrm{NCF}}^{(s)}.\end{aligned}\tag{7}$$

We then select the index $\widehat{b}^{(s)}$ that maximizes the standardized measure without cross-fitting, $\widehat{T}_{b,\mathrm{NCF}}^{(s)}$, in absolute value, that is, $\widehat{b}^{(s)}=\arg\max_{b\in\{1,\ldots,B\}}\left|\widehat{T}_{b,\mathrm{NCF}}^{(s)}\right|$. We take the measure with cross-fitting, $\widehat{T}_{\widehat{b}^{(s)},\mathrm{CF}}^{(s)}$, under the selected $\widehat{b}^{(s)}$, as our final test statistic.

We make a few remarks. First, we use the cross-fitting measure to construct the test statistic $\widehat{T}_{\widehat{b}^{(s)},\mathrm{CF}}^{(s)}$. This enables us to derive its limiting distribution more easily. Specifically, conditional on the data in $\mathcal{I}_s$, for each $b=1,\ldots,B$, $\widehat{T}_{b,\mathrm{CF}}^{(s)}$ converges

in distribution to standard normal under the null. Since $\widehat{b}^{(s)}$ is determined by $\widehat{T}^{(s)}_{b,\mathrm{NCF}}$, the index $\widehat{b}^{(s)}$ depends solely on the data in $\mathcal{I}_s$. Consequently, conditional on the data in $\mathcal{I}_s$, $\widehat{T}^{(s)}_{\widehat{b}^{(s)},\mathrm{CF}}$ converges in distribution to standard normal under the null as well. By contrast, the limiting distribution of the no-cross-fitting measure $\widehat{T}^{(s)}_{\widehat{b}^{(s)},\mathrm{NCF}}$ is unclear, due to the complicated dependence between $\widehat{b}^{(s)}$ and $\widehat{T}^{(s)}_{b,\mathrm{NCF}}$.

Second, we use the no-cross-fitting measure to select the index $\widehat{b}^{(s)}$. As we show in Section 4, when the estimated conditional mean function and the distributional generator belong to the VC type class (Chernozhukov, Chetverikov, and Kato 2014, Definition 2.1), the index $\widehat{b}^{(s)}$ that maximizes the no-cross-fitting measure $\{\widehat{T}^{(s)}_{b,\mathrm{NCF}}\}$ asymptotically maximizes the cross-fitting measure $\{\widehat{T}^{(s)}_{b,\mathrm{CF}}\}$ as well. This choice of the index $\widehat{b}^{(s)}$ is to maximize the power of the resulting test.

Finally, the random binary data splitting may introduce some sampling uncertainty. This issue is mitigated in our test, since we construct two test statistics based on both data subsets, then combine them to derive the final decision rule. One may also consider the multiple binary-splits idea of Meinshausen, Meier, and Bühlmann (2009), or the multi-split idea of Romano and DiCiccio (2019). We discuss a multiple binary-splits version of our test in Section S2.2 of the Appendix.

3.3. DAG Structural Learning

We next discuss the three key learning components (a) to (c) of our proposed test. The first is to estimate the DAG $\mathcal{G}$ associated with $X = (X_1, \ldots, X_d)^\top$, and to construct $\mathcal{M}$. In our implementation, we employ the neural structural learning method of Zheng et al. (2020). Other methods, for example, Yu et al. (2019); Zhu, Ng, and Chen (2020), can be used as well.

Consider a multilayer perceptron (MLP) with L hidden layers and an activation function σ:

$$\begin{aligned}&\mathrm{MLP}\left(u; A^{(1)}, b^{(1)}, \ldots, A^{(L)}, b^{(L)}\right) \\ &= A^{(L)}\sigma\left\{\cdots A^{(2)}\sigma\left(A^{(1)}\mu + b^{(1)}\right)\cdots + b^{(L-1)}\right\} + b^{(L)},\end{aligned} \tag{8}$$

where $u \in \mathbb{R}^{m_0}$ is the input signal of the MLP, $A^{(s)} \in \mathbb{R}^{m_\ell \times m_{\ell-1}}$, $b^{(s)} \in \mathbb{R}^{m_\ell}$ are the parameters that produce the linear transformation of the $(\ell-1)$th layer, the output is a scalar with $m_L = 1$, and there are m_ℓ nodes at layer ℓ, $\ell = 0, \ldots, L$. See Figure 1(b) for a graphical illustration.

We employ MLP to approximate the functions f_j's in our DAG model (1). In our theoretical analysis, we focus on the setting where f_j's are a set of continuous functions. Meanwhile, we may also consider a family of piecewise smooth functions (Imaizumi and Fukumizu 2019) for f_j's. In both cases, neural networks models such as MLP can consistently estimate f_j's. Let $\theta_j = \left\{A_j^{(\ell)}, b_j^{(\ell)} : 1 \le \ell \le L\right\}$ collect all the parameters for the jth MLP that approximates f_j, and let $\theta = \{\theta_j\}_{j=1}^d$. Accordingly, θ uniquely determines a graph structure, that is, how the variables are dependent to each other in the graph. We call this structure the graph induced by θ, and denote it by $\mathcal{G}(\theta)$. For each half of the data, $s = 1, 2$, we estimate the DAG via

$$\min_\theta \sum_{i\in\mathcal{I}_s}\sum_{t,j}\left\{\mathbb{X}_{i,t,j} - \mathrm{MLP}(\mathbb{X}_{i,t};\theta_j)\right\}^2, \quad \text{subject to } \mathcal{G}(\theta) \text{ is a DAG.}$$

This optimization, however, is challenging to solve, mainly due to the fact that the search space scales super-exponentially with the dimension d. To resolve this issue, Zheng et al. (2020) proposed a novel characterization of the acyclic constraint, and showed that the DAG constraint can be represented by $\mathrm{trace}[\exp\{W(\theta)\circ W(\theta)\}] = d$, where $\circ$ denotes the Hadamard product, $\exp(W)$ is the matrix exponential of W, $\mathrm{trace}(W)$ is the trace of W, and $W(\theta)$ is a $d \times d$ matrix whose (k,j)th entry equals the Euclidean norm of the kth column of $A_j^{(1)}$. Based on this characterization, the above optimization problem becomes,

$$\begin{aligned}\min_\theta \sum_{j=1}^d \left[\sum_{i\in\mathcal{I}_s}\sum_{t=1}^{T_i}\left\{\mathbb{X}_{i,t,j} - \mathrm{MLP}(\mathbb{X}_{i,t};\theta_j)\right\}^2 + \lambda n_s \left\|A_j^{(1)}\right\|_{1,1}\right], \\ \text{subject to } \ \mathrm{trace}[\exp\{W(\theta)\circ W(\theta)\}] = d,\end{aligned} \tag{9}$$

where $n_s = \sum_{i\in\mathcal{I}_s} T_i$ is the number of observations in $\mathcal{I}_s$, $\left\|A_j^{(1)}\right\|_{1,1}$ is the sum of all elements in $A_j^{(1)}$ in absolute values, and $\lambda > 0$ is a sparsity tuning parameter. Note that the sparsity penalization is placed only on $A_j^{(1)}$, since this is the only layer that determines the sparsity of the input variables $X_1, \ldots, X_d$. This new optimization problem in (9) can be efficiently solved using the augmented Lagrangian method (Zheng et al. 2020).

Let $\widehat{\mathcal{G}}^{(s)}$ denote the estimated graph, and $\widehat{\mathrm{AC}}_j$ and $\widehat{\mathrm{PA}}_j$ denote the corresponding estimated set of ancestors and parents of X_j, respectively. If $k \notin \widehat{\mathrm{AC}}_j^{(s)}$, then it follows from $\mathrm{PA}_j \subseteq \widehat{\mathrm{AC}}_j^{(s)}$ that $k \notin \mathrm{PA}_j$. Consequently, we simply set the corresponding p-value $p^{(s)}(j,k) = 1$. Our subsequent testing procedure is to focus on the case where $k \in \widehat{\mathrm{AC}}_j^{(s)}$, and we set $\mathcal{M}^{(s)} = \widehat{\mathrm{AC}}_j^{(s)} - \{k\}$. We also remark that, to establish the consistency of our test, we only require $\mathbb{P}(\mathrm{PA}_j \subseteq \widehat{\mathrm{AC}}_j^{(s)} \subseteq \mathrm{DS}_j^c - \{j\}) \to 1$, where DS_j^c denotes the complement of the set DS_j. This essentially requires the order of the DAG to be consistently estimated. We later show in Section S2.1 that this condition is satisfied when using the method of Zheng et al. (2020). Meanwhile, this order consistency is much weaker than requiring the DAG estimator $\widehat{\mathcal{G}}^{(s)}$ to be selection consistent, that is, $\mathbb{P}(\mathrm{PA}_j = \widehat{\mathrm{PA}}_j) \to 1$, or to satisfy sure screening, that is, $\mathbb{P}(\mathrm{PA}_j \subseteq \widehat{\mathrm{PA}}_j) \to 1$.

3.4. Supervised Learning

The second key component of our test is to learn the conditional mean $g^{(s)}(x) = \mathbb{E}\left(X_j | X_{\mathcal{M}^{(s)}} = x\right)$. This is essentially a regression problem, and there are many choices, for example, boosting, random forests, or neural networks. In our implementation, we use the MLP again, by seeking

$$\min_{\theta_j} \sum_{i\in\mathcal{I}_s}\sum_{t=1}^{T_i}\left\{\mathbb{X}_{i,t,j} - \mathrm{MLP}\left(\mathbb{X}_{i,t,\mathcal{M}^{(s)}};\theta_j\right)\right\}^2, \tag{10}$$

where the learner $\mathrm{MLP}(\cdot)$ is as defined in (8). The optimization problem in (10) can be solved using a stochastic gradient descent algorithm, or the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (Byrd et al. 1995).

3.5. Generative Adversarial Learning

The third key component of our test is to use GANs to learn a generator $\mathbb{G}^{(s)}(\cdot,\cdot)$, which generates a set of pseudo samples that have a similar distribution as the training samples. More accurately, in our setting, we learn the generator $\mathbb{G}(\cdot,\cdot)$ that takes $\mathbb{X}_{i,t,\mathcal{M}-\{k\}}$ and a set of multivariate Gaussian noise vectors as the input, and the output are a set of pseudo samples $\widetilde{\mathbb{X}}_{i,t,k}^{(s,m)}$. We train the generator such that the divergence between the conditional distribution of $\mathbb{X}_{i,t,k}$ given $\mathbb{X}_{i,t,\mathcal{M}-\{k\}}$ and that of $\widetilde{\mathbb{X}}_{i,t,k}^{(s,m)}$ given $\mathbb{X}_{i,t,\mathcal{M}-\{k\}}$ is minimized.

More specifically, we adopt Genevay, Peyré, and Cuturi (2018) to learn the generator $\mathbb{G}^{(s)}$, by optimizing

$$\min_{\mathbb{G}} \max_{c} \widetilde{\mathcal{D}}_{c,\rho}(\mu,\nu), \tag{11}$$

where μ and ν denote the joint distribution of $\left(\mathbb{X}_{i,t,k}, \mathbb{X}_{i,t,\mathcal{M}^{(s)}}\right)$ and $\left(\widetilde{\mathbb{X}}_{i,t,k}^{(s,m)}, \mathbb{X}_{i,t,\mathcal{M}^{(s)}}\right)$, respectively, and $\widetilde{\mathcal{D}}_{c,\rho}$ is the Sinkhorn loss function between two probability measures. The loss $\widetilde{\mathcal{D}}_{c,\rho}$ is with respect to a cost function c and a regularization parameter $\rho > 0$,

$$\widetilde{\mathcal{D}}_{c,\rho}(\mu,\nu) = 2\mathcal{D}_{c,\rho}(\mu,\nu) - \mathcal{D}_{c,\rho}(\mu,\mu) - \mathcal{D}_{c,\rho}(\nu,\nu),$$

$$\mathcal{D}_{c,\rho}(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int_{x,y} \left\{c(x,y) - \rho H(\pi|\mu\otimes\nu)\right\}\pi(dx,dy),$$

where $\Pi(\mu,\nu)$ is a set containing all probability measures π whose marginal distributions correspond to μ and ν, H is the Kullback-Leibler divergence, and $\mu\otimes\nu$ is the product measure of μ and ν. When $\rho = 0$, $\mathcal{D}_{c,0}(\mu,\nu)$ measures the optimal transport of μ into ν with respect to the cost function $c(\cdot,\cdot)$ (Cuturi 2013). When $\rho \neq 0$, an entropic regularization is added to this optimal transport. As such, the objective function $\widetilde{\mathcal{D}}_{c,\rho}$ in (11) is a regularized optimal transport metric, where the regularization is to facilitate the computation, so that $\widetilde{\mathcal{D}}_{c,\rho}$ can be efficiently evaluated. Intuitively, the closer the two conditional distributions, the smaller the Sinkhorn loss. Therefore, maximizing $\widetilde{\mathcal{D}}_{c,\rho}$ with respect to the cost c learns a discriminator that can better discriminate μ and ν. On the other hand, minimizing the maximum cost with respect to the generator $\mathbb{G}$ makes the conditional distribution of $\widetilde{\mathbb{X}}_{i,t,k}^{(s,m)}$ given $\mathbb{X}_{i,t,\mathcal{M}^{(s)}}$ closer to that of $\mathbb{X}_{i,t,k}$ given $\mathbb{X}_{i,t,\mathcal{M}^{(s)}}$. This yields the minimax formulation in (11). In our implementation, we approximate the cost function c and the generator based on MLP (8). We approximate the distributions $\mu_{j,k}$ and $\nu_{j,k}$ in (11) by the empirical distributions of the data samples. We update the parameters in GANs by the Adam algorithm (Kingma and Ba 2015).

We again make a few remarks. First, we choose the Gaussian noise as the input for GANs. We have found the performance of the generator is not overly sensitive to the choice of the distribution of the input noise. We present more discussion and some additional numerical results in Section S2.3 of the Appendix. Besides, we choose GANs based on the Sinkhorn divergence loss to mitigate the potential bias of traditional GANs. Moreover, in addition to GANs, other deep generative learning approaches such as variational auto-encoders (Kingma and Welling 2013) are equally applicable here. Second, we note that, based on the estimated conditional distribution from GANs, one can derive the joint distribution of all variables, then infer the corresponding DAG structure. However, this may be computationally inefficient, due to the huge number of conditional dependence relations that must be learned. Finally, we note that, an alternative approach for this step is to separately apply a supervised learning method B times to estimate $\mathbb{E}\left\{h_b\left(X_k, X_{\mathcal{M}-\{k\}}\right)|X_{\mathcal{M}-\{k\}}\right\}$, for $b = 1,\ldots,B$. Nevertheless, when B is large, and in our implementation, $B = 2000$, this approach is computationally very expensive. Therefore, we choose the generative learning approach for this step.

4. Bidirectional Theory

In this section, we establish the asymptotic size and power of the proposed test. As a by-product, we also derive the oracle property of the DAG estimator produced by (9), which is needed to guarantee the validity of the test. In the interest of space, we report that result in Section S2.1 of the Appendix. To simplify the theoretical analysis, we assume $T_1 = \cdots = T_n = T$. All the asymptotic results are derived when either the number of subjects N, or the number of time points T, diverges to infinity. Such results are new, provide useful theoretical guarantees for different types of applications, and are referred as the bidirectional theory.

We begin with a set of regularity conditions needed for the asymptotic consistency.

(C1) With probability approaching one, $\mathrm{PA}_j \subseteq \widehat{\mathrm{AC}}_j^{(s)} \subseteq \mathrm{DS}_j^c - \{j\}$.

(C2) Suppose $\mathbb{E}\left|g^{(s)}\left(X_{\mathcal{M}^{(s)}}\right) - \widehat{g}^{(s)}\left(X_{\mathcal{M}^{(s)}}\right)\right|^2 = O\left\{(NT)^{-2\kappa_1}\right\}$ for some constant $\kappa_1 > 0$, and $\widehat{g}^{(s)}$ is uniformly bounded almost surely. Suppose $\mathbb{E}\sup_{\widetilde{B}\in\mathcal{B}}\left|\mathbb{P}\left\{X_k \in \widetilde{B}|X_{\mathcal{M}^{(s)}}\right\} - \mathbb{P}\left\{\mathbb{G}^{(s)}\left(X_{\mathcal{M}^{(s)}}, Z_{j,k}^{(m)}\right) \in \widetilde{B}|\, X_{\mathcal{M}^{(s)}}\right\}\right|^2 = O\left\{(NT)^{-2\kappa_2}\right\}$ for some constant $\kappa_2 > 0$, where $\mathcal{B}$ denotes the Borel algebra on $\mathbb{R}$. Suppose $\kappa_1 + \kappa_2 > 1/2$.

(C3) The random process $\{\mathbb{X}_{i,t}\}_{t\geq 0}$ is β-mixing if T diverges to infinity. The β-mixing coefficients $\{\beta(q)\}_q$ satisfy that $\sum_q q^{\kappa_3}\beta(q) < +\infty$ for some constant $\kappa_3 > 0$. Here, $\beta(q)$ denotes the β-mixing coefficient at lag q, which measures the time dependence between the set of variables $\{\mathbb{X}_{i,j}\}_{j\leq t}$ and $\{\mathbb{X}_{i,j}\}_{j\geq t+q}$.

(C4) Suppose the number of observations K in the batched standard error estimators $\widehat{\sigma}_{b,\mathrm{CF}}^{(s)}$ and $\widehat{\sigma}_{b,\mathrm{NCF}}^{(s)}$ satisfies that, $K = T$ if T is bounded, and $T^{(1+\kappa_3)^{-1}} \ll K \ll NT$ otherwise.

Condition (C1) concerns about the step of structural learning of DAG, which essentially requires the order of the DAG can be consistently estimated. We first remark that, this order consistency is much weaker than the selection consistency. In other words, we only require a reasonably good initial DAG estimator that is order consistent, which is much easier to obtain than a DAG estimator that is selection consistent. In Section S2.1, we show that (C1) holds when (9) is employed to estimate the DAG. Second, (C1) may not be a necessary condition to ensure the Type-I error control. We next give two examples, where (C1) does not hold, but our proposed test can still control the Type-I error. Moreover, in our simulation examples in Section 5, (C1)

does not always hold either. We report the percentage of times out of 500 data replications when (C1) holds for some selected nodes in Section S2.4 of the Appendix. Nevertheless, our test still manages to achieve a competitive empirical performance. On the other hand, we keep (C1) in its current form, as it helps simplify the proof considerably.

Example 5 (missing parents). We first consider an example where $\widehat{\mathrm{AC}}_j^{(s)}$ misses some nodes in PA_j. The proposed test remains valid as long as these nodes have weak effects on X_j and X_k. More specifically, consider the five-variable example as illustrated in Figure 1(c). Our goal is to test whether there is a directed link from X_3 to X_4. Then $\mathrm{PA}_j \subseteq \widehat{\mathrm{AC}}_j^{(s)}$ requires that $\{1,2\} \subseteq \widehat{\mathrm{AC}}_4^{(s)}$. Suppose X_1 has a weak effect on X_4, so that X_1 is not included in $\widehat{\mathrm{AC}}_4^{(s)}$. Suppose $|\mathbb{E}(X_4|X_1,X_2) - \mathbb{E}(X_4|X_2)|^2 = O\{(NT)^{-2\kappa_1^*}\}$, for some $\kappa_1^* \geq \kappa_1$. When $\mathbb{E}\sup_{\widetilde{B}\in\mathcal{B}} |\mathbb{P}(X_3 \in \widetilde{B}|X_2) - \mathbb{P}(X_3 \in \widetilde{B}|X_1,X_2)| = O\{(NT)^{-2\kappa_2^*}\}$, for some $\kappa_2^* \geq \kappa_2$, under (C2)-(C4), the estimated conditional mean function and the distributional generator would converge to $\mathbb{E}(X_4|X_1,X_2)$ and $\mathbb{P}_{X_3|X_1,X_2}$ at the rate of $(NT)^{-\kappa_1}$ and $(NT)^{-\kappa_2}$, respectively. As such, the proposed test still works as if X_1 were included in $\widehat{\mathrm{AC}}_4^{(s)}$.

Example 6 (including descendants). We next consider an example where $\widehat{\mathrm{AC}}_j^{(s)}$ includes some nodes in DS_j. The proposed test remains valid as long as none of these nodes is a descendant of X_k, or has a common descendant with X_k. In this case, X_k and X_j are d-separated given $\widehat{\mathrm{AC}}_j^{(s)}$, as none of those falsely included nodes is a collider on any path between X_j and X_k; see the definition of d-separation and collider in Pearl (2009). As d-separation implies conditional independence, the proposed test is still able to control the Type-I error. For the example in Figure 1(c), when $\{5\} \in \widehat{\mathrm{AC}}_4^{(s)}$, (C1) is violated. However, when X_3 does not have affect X_5, the proposed test remains valid.

Condition (C2) concerns about the steps of learning the conditional mean function and the distribution generator. It requires the squared prediction loss of the supervised learner of the conditional mean, and the squared total variation norm between the conditional distributions of the observed and pseudo samples to satisfy some convergence rate, κ_1 and κ_2, respectively. We note that both estimators are nonparametric, and as such, both κ_1 and κ_2 can be slower than the parametric rate of 1/2. However, (C2) only requires that $\kappa_1 + \kappa_2 > 1/2$. This is relatively easy to achieve when using the multilayer perceptron models and GANs, whose convergence rates have been established (see e.g., Schmidt-Hieber 2017; Liang 2018; Bauer and Kohler 2019; Chen et al. 2020; Farrell, Liang, and Misra 2021). Moreover, we remark that, it is possible to further relax the requirement of $\kappa_1+\kappa_2 > 1/2$ to $\kappa_1, \kappa_2 > 0$, by using the theory of higher order influence functions (Robins et al. 2017). However, the corresponding estimators would be considerably much more complicated, and thus we do not pursue those in this article.

Condition (C3) characterizes the dependence of the data observations over time, and is commonly imposed in the time series literature (Bradley 2005). We also note that, (C3) is *not* needed when T is bounded but N diverges to infinity. Condition (C4) guarantees the consistency of the batched standard error estimators $\widehat{\sigma}_{b,\mathrm{CF}}^{(s)}$ and $\widehat{\sigma}_{b,\mathrm{NCF}}^{(s)}$, and is easily satisfied, since K is a parameter we specify. When T is bounded and is relatively small compared to a large sample size N, we can simply set $K = T$, that is, treating the entire time series as one batch.

We next establish the asymptotic size of the propose testing procedure.

Theorem 2 (Size). Suppose model (1), and conditions (C1)–(C4) hold. Suppose $\min_b NT\ \mathrm{var}\Big(\widehat{I}_{b,\mathrm{CF}}^{(s)}|\{\mathbb{X}_{i,t}\}_{i\in\mathcal{I}_s,1\leq t\leq T}\Big) \geq \kappa_4$ for some constant $\kappa_4 > 0$. If the constants $\kappa_1, \kappa_2, \kappa_3$ satisfy that $\kappa_3 > \max[\{2\min(\kappa_1,\kappa_2)\}^{-1} - 1, 2]$, then, as either N or $T \to \infty$,

(a) The test statistic $\widehat{T}_{\widehat{b}^{(s)},\mathrm{CF}}^{(s)} \xrightarrow{d} \mathrm{Normal}(0,1)$ under $H_0(j,k)$.
(b) The p-value satisfies that $\mathbb{P}\{p(j,k) \leq \alpha\} \leq \alpha + o(1)$, for any nominal level $0 < \alpha < 1$.

To establish the asymptotic size of the test, we require $\beta(q)$ to decay at a polynomial rate with respect to q. Such a condition holds for many common time series models (see, e.g., McDonald, Shalizi, and Schervish 2015). We also require a minimum variance condition, which automatically holds when the conditional variance of $h_b^{(s)}\big(X_k, X_{\mathcal{M}^{(s)}}\big) - \mathbb{E}\big\{h_b^{(s)}\big(X_k, X_{\mathcal{M}^{(s)}}\big)|X_{\mathcal{M}^{(s)}}\big\}$ given $X_{\mathcal{M}^{(s)}}$ is bounded away from zero. Under these conditions, we establish the asymptotic normality of the test statistic $\widehat{T}_{\widehat{b}^{(s)},\mathrm{CF}}^{(s)}$, which further implies that the p-value $p^{(s)}(j,k)$ converges to a uniform distribution on $[0,1]$. By Bonferroni's inequality, $p(j,k)$ is a valid p-value, and consequently, the proposed test achieves a valid control of Type-I error.

Next, we study the asymptotic power of the test. We introduce a quantity to characterize the degree to which the alternative hypothesis deviates from the null for a given function class $\mathbb{H}$: $\Delta(\mathbb{H}) = \min_{\mathcal{M}} \sup_{h\in\mathbb{H}} |I(j,k|\mathcal{M};h)|$, where the minimum is taken over all subsets $\mathcal{M}$ that satisfy the conditions in Proposition 1. When $\mathbb{H}$ is taken over the class of characteristic functions of $(X_k, X_{\mathcal{M}})$, we have $\Delta(\mathbb{H}) > 0$. We also need the concept of the VC type class (Chernozhukov, Chetverikov, and Kato 2014, Definition 2.1); see Section S3.4 of the Appendix. To simplify the analysis, we suppose X_j is bounded, and without loss of generality, its support is $[0,1]$.

Theorem 3 (Power). Suppose the conditions in Theorem 2 hold, and the β-mixing coefficient $\beta(q)$ in (C3) satisfies that $\beta(q) = O(\kappa_5^q)$ for some constant $0 < \kappa_5 < 1$ when T diverges. Suppose $\Delta(\mathbb{H}) \gg (NT)^{-1/2}\log(NT)$ under $H_1(j,k)$. Suppose, with probability tending to one, $\widehat{g}^{(s)}$ and $\mathbb{G}^{(s)}$ belong to the class of VC type functions with bounded envelope functions and the bounded VC indices no greater than $O\{(NT)^{\min(2\kappa_1,2\kappa_2,1/2)}\}$, $s = 1,2$. If the number of transformation functions $B = \kappa_6(NT)^{\kappa_7}$ for some constants $\kappa_6 > 0, \kappa_7 \geq 1/2$, then, as either N or $T \to \infty$, $p(j,k) \xrightarrow{p} 0$ under $H_1(j,k)$.

To establish the asymptotic power of the test, we require the function $\widehat{g}^{(s)}$ and the generator $\mathbb{G}^{(s)}$ to both belong to the VC type class. This is to help establish the concentration inequalities for the measure $\widehat{I}_{b,\mathrm{NCF}}^{(s)}$ without cross-fitting. This condition automatically holds in our implementation where the MLP is

used to model both (Farrell, Liang, and Misra 2021). We have also strengthened the requirement on $\beta(q)$, so that it decays exponentially with respect to q. This is to ensure the $\sqrt{NT}$-consistency of the proposed test when $T \to \infty$. This condition holds when the process $\{\mathbb{X}_{i,t}\}_{t\geq 0}$ forms a recurrent Markov chain with a finite state space. It also holds for more general state space Markov chains (see, e.g., Bradley 2005, sec. 3). Under these conditions, Theorem 3 shows that our proposed test is consistent against some local alternatives that are $\sqrt{NT}$-consistent to the null up to some logarithmic term.

We remark that, Theorems 2 and 3 show that the proposed test controls the Type-I error and achieves a parametric power guarantee, even though we estimate the three key components, the DAG structure, the conditional mean, and the distribution generator, all using fully nonparametric methods. This is achieved mainly due to the fact that our test statistic $\widehat{T}^{(s)}_{\widehat{b}^{(s)},\mathrm{CF}}$ is doubly robust, in that it is consistent as long as either the conditional mean or the distribution generator is correctly specified. Together with the Neyman orthogonality of the estimating equation, we show that the bias can be represented as a product of the difference between the two nonparametric estimators and their oracle values; see Step 3 of the proof of Theorem 2 in Section S3.3 of the Appendix. Consequently, as long as $\kappa_1+\kappa_2 > 1/2$, the test statistic converges at a parametric rate, and the test has a parametric power guarantee.

We also remark that, in our theory, the dimension d of the DAG is allowed to diverge to infinity with the sample size. Note that there is no explicit specification on d in the statements of Theorems 2 and 3. It is implicitly imposed due to the requirement that $\kappa_1+\kappa_2 > 1/2$, as the convergence rates would become slower as the dimension d increases.

5. Simulations

In this section, we examine the finite-sample performance of the proposed testing procedure.

We begin with a discussion of some implementation details. Our test employs three neural networks-based learners, which involve numerous tuning parameters. Many of these parameters are common, for example, the number of hidden layers and hidden nodes, the activation function, batch size, and epoch size, and we set them at the typical values recommended in the literature. For the DAG learning step, one tuning parameter is the sparsity parameter λ in (9). Following Zheng et al. (2020), we fix $\lambda = 0.025$ in our implementation to speed up the computation. We have also experimented with a number of values of λ and find the results are not overly sensitive. It can also be tuned via cross-validation. For the supervised learning step, we employ the multilayer perceptron regressor implementation of Pedregosa et al. (2011). For the GANs training step, we follow the implementation of Genevay, Peyré, and Cuturi (2018). There are three additional parameters associated with our test, including the number of transformation functions B, the number of pseudo samples M, and the number of observations K in the batched standard error estimators. We have found that the results are not sensitive to the choice of M and K, and we fix $M = 100$ and $K = 20$. For B, a larger value generally improves the power of the test, but also increases the computational cost. In our implementation, we set $B = 2000$, which achieves a reasonable balance between the test accuracy and the computational cost.

We compare the proposed test with two alternative solutions, the double regression-based test (DRT) as outlined in Section 2.2, and the constrained likelihood ratio test (LRT) proposed by Li, Shen, and Pan (2020) for linear DAGs. The implementation of DRT is similar to our proposed method. The main difference lies in that DRT uses the MLP regressor to first estimate the conditional mean function $\mathbb{E}(X_k|X_{\mathcal{M}^{(j)}_{j,k}})$ in Step 4, then plugs in this estimate to construct the test statistic in Step 5, with $B = 1$ and $h_1^{(s)}(X_k, X_{\mathcal{M}^{(j)}_{j,k}}) = X_k$.

We consider the following nonlinear DAG model,

$$X_{t,j} = \sum_{\substack{k_1,k_2\in \mathrm{PA}_j \\ k_1\leq k_2}} c_{j,k_1,k_2} f^{(1)}_{j,k_1,k_2}(X_{t,k_1}) f^{(2)}_{j,k_1,k_2}(X_{t,k_2}) + \sum_{k_3\in \mathrm{PA}_j} c_{j,k_3} f^{(3)}_{j,k_3}(X_{t,k_3}) + \varepsilon_{t,j}. \tag{12}$$

The data generation follows that of Zhu, Ng, and Chen (2020). Specifically, $f^{(1)}_{j,k_1,k_2}$, $f^{(2)}_{j,k_1,k_2}$, and $f^{(3)}_{j,k_3}$ in (12) are randomly set to be sine or cosine function with equal probability, whereas c_{j,k_1,k_2} and c_{j,k_3} are randomly generated from uniform $[0.5\delta, 1.5\delta]$ or $[-1.5\delta, -0.5\delta]$ with an equal probability, where $\delta > 0$ denotes some constant that controls the signal strength. The error $\varepsilon_{t,j}$ is an AR(1) process with the autoregressive coefficient equal to 0.5 and a standard normal white noise. The DAG structure is determined by a $d \times d$ lower triangular binary adjacency matrix, in which each entry is randomly sampled from a Bernoulli distribution with probability ζ. We vary four sets of key parameters in our simulations: (a) the number of subjects N from $\{10, 20, 40\}$; (b) the number of time points T from $\{50, 100, 200\}$; (c) the signal strength δ from $\{0.5, 1, 2\}$, and (d) the dimension d and the Bernoulli probability ζ from $(d, \zeta) = \{(50, 0.10), (100, 0.04), (150, 0.02)\}$. When we vary one set of the parameters, we keep the rest fixed at their default values of $N = 20, T = 100, \delta = 1, d = 50, \zeta = 0.10$.

For each scenario, we randomly sample 100 pairs of nodes where the null hypothesis holds, and another 100 pairs of nodes where the alternative hypothesis holds. We then apply the proposed test to these pairs, and record the empirical size and power of the test, that is, the percentage of the times out of 200 data replications when the p-value is smaller than the nominal level $\alpha = 0.05$. Figure 2 shows the boxplots of the empirical size for the pairs when the null holds, and Figure 3 shows the boxplots of the empirical power for the pairs when the alternative holds. We further report the difference of the powers of SUGAR and DRT in Figure S2 in Section S2.5 of the Appendix. We do not report the power of LRT, because it fails to control the Type-I error, and thus its empirical power becomes meaningless. We make the following observations from these plots. In terms of the empirical size, both SUGAR and DRT manage to control the Type-I error, but LRT does not. The reason is that LRT requires the graph to have a linear structure and the samples to be independent, but none is satisfied in our simulation model. On the other hand, in terms of the empirical power, SUGAR achieves generally a

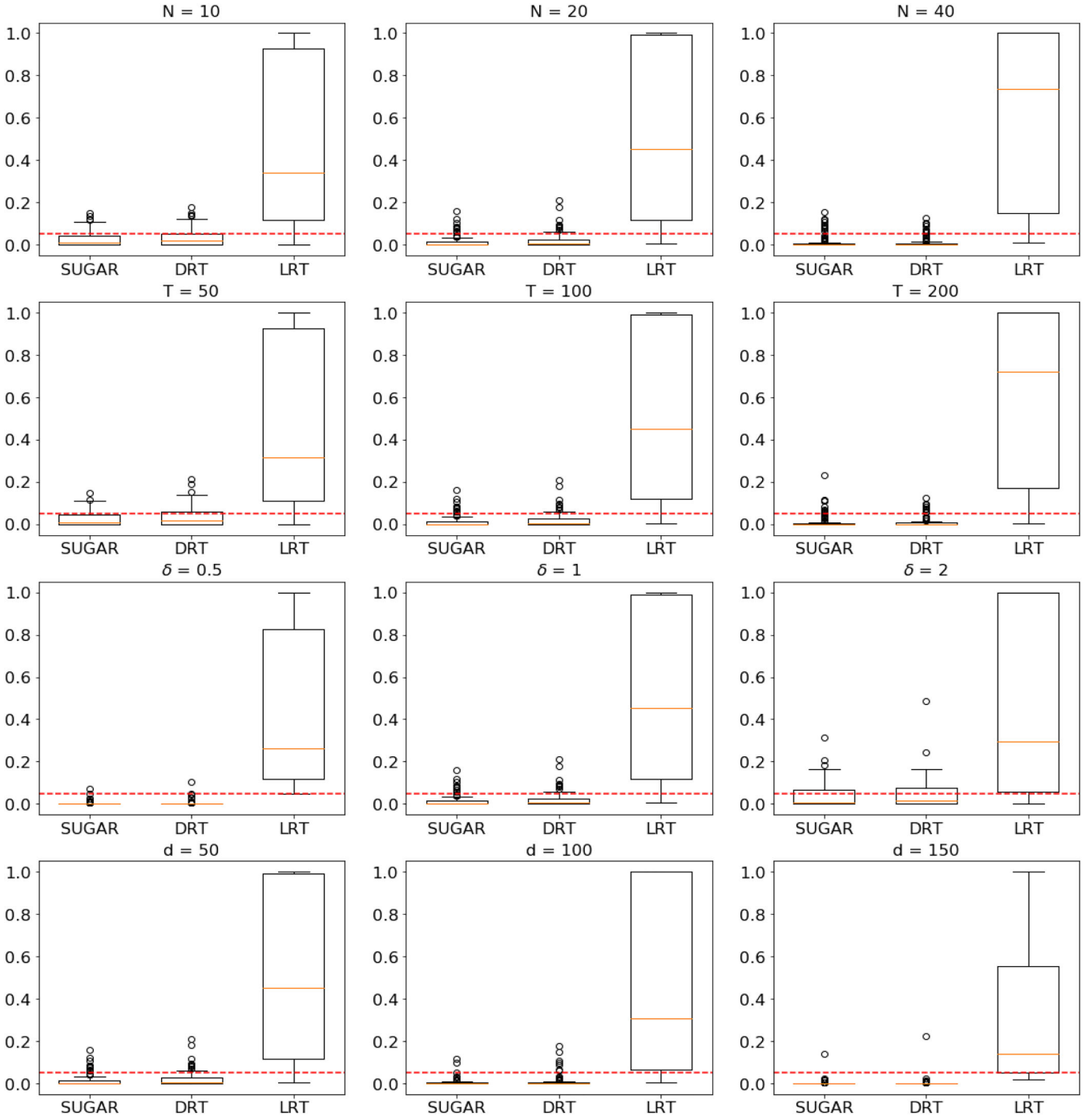

Figure 2. The boxplots of the empirical size of three methods: our proposed test (SUGAR), the double regression-based test (DRT), and the constrained likelihood ratio test (LRT), under four sets of varying parameters: first row $N = \{10, 20, 40\}$, second row $T = \{50, 100, 200\}$, third row $\delta = \{0.5, 1, 2\}$, and fourth row $(d, \zeta) = \{(50, 0.10), (100, 0.04), (150, 0.02)\}$.

higher power than DRT, over 75% of the times in all scenarios as seen from Figure S2. Finally, as the key model parameters vary, the power of both SUGAR and DRT increases as the number of subjects N, or the number of time points T increases, since more data information becomes available, and the power of both tests decreases as the dimension d increases, since the graph becomes bigger and the problem more challenging. Meanwhile, the power of SUGAR increases as the signal strength δ increases, but that of DRT is not monotonic with respect to δ, because DRT is not guaranteed to be consistent in general, as we have commented earlier.

In terms of the computational time, our testing procedure consists of two main parts: the DAG estimation in Step 2 of Algorithm 1, and the rest in Steps 3–6. The DAG estimation is the most time consuming step, but it only needs to be learnt once for all pairs of edges in the graph. We implemented the DAG estimation step on the NVIDIA Tesla T4 GPU, and it took about 5–20 min when d ranges from 50 to 150 for one data replication. We implemented the rest of the testing procedure on the N1 standard CPU, and it took about 2 min for one data replication. A Python implementation of our method is available at *https://github.com/yunzhe-zhou/SUGAR*.

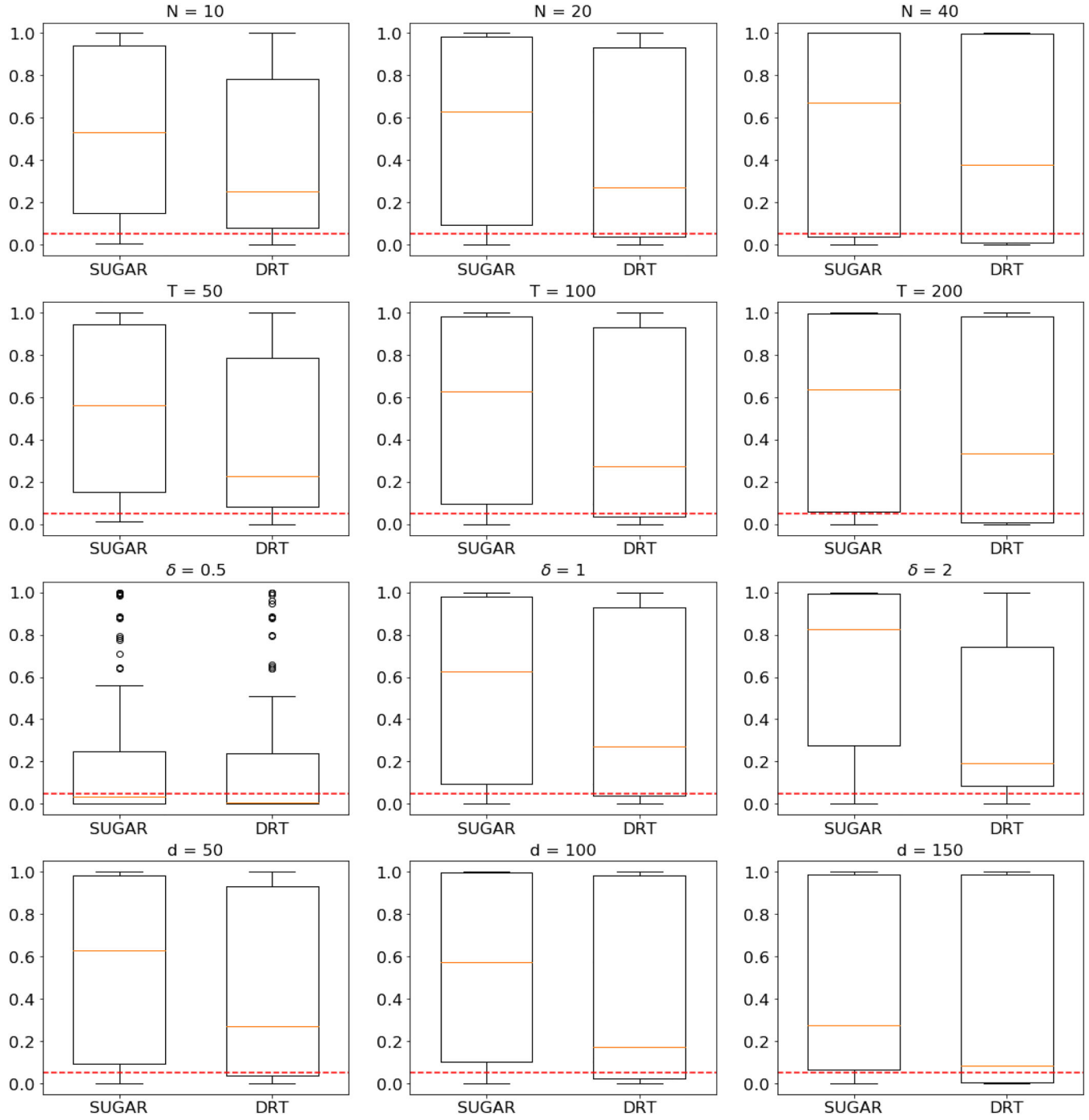

Figure 3. The boxplots of the empirical power of two methods: our proposed test (SUGAR), and the double regression-based test (DRT), under four sets of varying parameters: first row $N = \{10, 20, 40\}$, second row $T = \{50, 100, 200\}$, third row $\delta = \{0.5, 1, 2\}$, and fourth row $(d, \zeta) = \{(50, 0.10), (100, 0.04), (150, 0.02)\}$.

6. Brain Effective Connectivity Analysis

We next illustrate our method with a brain effective connectivity analysis of task-evoked functional magnetic resonance imaging (fMRI) data. The brain is a highly interconnected dynamic system, and it is of great interest to understand the relations among different brain regions through fMRI, which measures synchronized blood oxygen level dependent brain signals. The dataset we analyze is part of the Human Connectome Project (HCP, Van Essen et al. 2013), whose overarching objective is to understand brain connectivity patterns of healthy adults. We study the fMRI scans of a group of individuals who undertook a story-math task. The task consisted of blocks of auditory stories and addition-subtraction calculations, and required the participant to answer a series of questions. An accuracy score was given at the end. We analyze two subsets of individuals with matching age and sex. One set consists of $N = 28$ individuals who scored below 65 out of 100, and the other set consists of $N = 28$ individuals who achieved the perfect score of 100. All fMRI scans have been preprocessed following the pipeline of Glasser et al. (2013) that summarized each fMRI scan as a matrix of time series. Each row is a time series with length $T = 316$, and there are 264 rows corresponding to 264 brain regions (Power

Table 1. The number of identified significant within-module and between-module connections of the four functional modules for the low-performance and high-performance groups.

	Auditory (13)		Default mode (58)		Visual (31)		Fronto-parietal (25)	
	Low	High	Low	High	Low	High	Low	High
Auditory (13)	20	17	0	0	0	1	2	0
Visual (31)	0	0	3	2	56	46	0	1
Fronto-parietal (25)	2	1	11	23	0	1	22	27

NOTE: The number of brain regions of each functional module is reported in the parenthesis.

et al. 2011). Those brain regions are further grouped into 14 functional modules (Smith et al. 2009). Each module possesses a relatively autonomous functionality, and complex tasks are believed to perform through coordinated collaborations among the modules. In our analysis, we concentrate on $d = 127$ brain regions from four functional modules: auditory, visual, frontoparietal task control, and default mode, which are generally believed to be involved in language processing and problem solving domains (Barch et al. 2013).

We apply the proposed test to the two datasets separately. We control the false discovery at 0.05 using the standard Benjamini-Hochberg procedure (Benjamini and Hochberg 1995). Table 1 reports the number of identified significant within-module and between-module connections. We first note that, we identify many more within-module connections than the between-module connections. The partition of the brain regions into the functional modules has been fully based on the biological knowledge, and our finding lends some numerical support to this partition. In addition, we identify more within-module connections for the frontoparietal task control module for the high-performance subjects than the low-performance subjects, while we have identified fewer within-module connections for the default mode and visual modules for the high-performance subjects. These findings generally agree with the neuroscience literature. Particularly, the frontoparietal network is known to be involved in sustained attention, complex problem solving and working memory (Menon 2011), and the high-performance group exhibits more active connections for this module. Meanwhile, the default mode network is more active during passive rest and mind-wandering, which usually involves remembering the past or envisioning the future rather than the task being performed (Van Praag et al. 2017), and the high-performance group exhibits fewer active connections for this module.

Supplementary Materials

Section A of the supplementary article discusses several extensions of the proposed test. Section B presents additional theoretical and numerical results. Section C gives the detailed proofs.

Acknowledgments

The authors wish to thank the Editor, the AE, and the reviewers for their constructive comments, which have led to a significant improvement of the earlier version of this article.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

Li's research was partially supported by NSF grant CIF-2102227, and NIH grants R01AG061303, and R01AG062542. Shi's research was partially supported by EPSRC grant EP/W014971/1.

References

Barch, D. M., Burgess, G. C., Harms, M. P., Petersen, S. E., Schlaggar, B. L., Corbetta, M., Glasser, M. F., Curtiss, S., Dixit, S., Feldt, C., Nolan, D., Bryant, E., Hartley, T., Footer, O., Bjork, J. M., Poldrack, R., Smith, S., Johansen-Berg, H., Snyder, A. Z., Van Essen, D. C., and WU-Minn HCP Consortium. (2013), "Function in the Human Connectome: Task-fMRI and Individual Differences in Behavior," *NeuroImage*, 80, 169–189. Mapping the Connectome. [13]

Bauer, B., and Kohler, M. (2019), "On Deep Learning as a Remedy for the Curse of Dimensionality in Nonparametric Regression," *The Annals of Statistics*, 47, 2261–2285. [2,9]

Benjamini, Y., and Hochberg, Y. (1995), "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society*, Series B, 57, 289–300. [13]

Bradley, R. C. (2005), "Basic Properties of Strong Mixing Conditions. A Survey and Some Open Questions," *Probability Survey*, 2, 107–144. Update of, and a supplement to, the 1986 original. [9,10]

Bühlmann, P., Peters, J., and Ernest, J. (2014), "CAM: Causal Additive Models, High-Dimensional Order Search and Penalized Regression," *The Annals of Statistics*, 42, 2526–2556. [3]

Bullmore, E., and Sporns, O. (2009), "Complex Brain Networks: Graph Theoretical Analysis of Structural and Functional Systems, *Nature Reviews. Neuroscience*, 10, 186–198. [5]

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995), "A Limited Memory Algorithm for Bound Constrained Optimization," *SIAM Journal on Scientific Computing*, 16, 1190–1208. [7]

Cai, T. T. (2017), "Global Testing and Large-Scale Multiple Testing for High-Dimensional Covariance Structures," *Annual Review of Statistics and Its Application*, 4, 423–446. [2]

Carlstein, E. (1986), "The Use of Subseries Values for Estimating the Variance of a General Statistic from a Stationary Sequence," *The Annals of Statistics*, 14, 1171–1179. [6]

Chen, M., Liao, W., Zha, H., and Zhao, T. (2020), "Statistical Guarantees of Generative Adversarial Networks for Distribution Estimation," arXiv preprint arXiv:2002.03938. [9]

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018), "Double/Debiased Machine Learning for Treatment and Structural Parameters," *The Econometrics Journal*, 21, C1–C68. [3]

Chernozhukov, V., Chetverikov, D., and Kato, K. (2014), "Gaussian Approximation of Suprema of Empirical Processes," *The Annals of Statistics*, 42, 1564–1597. [7,9]

Chickering, D. M., Heckerman, D., and Meek, C. (2004), "Large-Sample Learning of Bayesian Networks is NP-Hard," *Journal of Machine Learning Research*, 5, 1287–1330. [1]

Cuturi, M. (2013), "Sinkhorn Distances: Lightspeed Computation of Optimal Transport," in *Advances in Neural Information Processing Systems*, pp. 2292–2300. [8]

Farrell, M. H., Liang, T., and Misra, S. (2021), "Deep Neural Networks for Estimation and Inference," *Econometrica*, 89, 181–213. [2,5,9,10]

Friston, K. J. (2011), "Functional and Effective Connectivity: A Review," *Brain Connectivity*, 1, 13–36. [1]

Garg, R., Cecchi, G., and Rao, R. (2011), "Full-Brain Auto-Regressive Modeling (FARM) Using fMRI," *NeuroImage*, 58, 416–441. [1]

Genevay, A., Peyré, G., and Cuturi, M. (2018), "Learning Generative Models with Sinkhorn Divergences," in *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. [5,8,10]

Glasser, M. F., Sotiropoulos, S. N., Anthony Wilson, J., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., Van Essen, D. C., Jenkinson, M., and WU-Minn HCP Consortium. (2013), "The Minimal Preprocessing Pipelines for the Human Connectome Project," *Neuroimage*, 80, 105–124. [12]

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014), "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680. [2,5]

Imaizumi, M., and Fukumizu, K. (2019), "Deep Neural Networks Learn Non-smooth Functions Effectively," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 869–878, PMLR. [7]

Janková, J., and van de Geer, S. (2019), "Inference in High-Dimensional Graphical Models," in *Handbook of Graphical Models*, Chapman & Hall/Chapman & Hall/CRC Handbooks of Modern Statistical Methods, eds. M. Maathuis, M. Drton, S. Lauritzen, and M. Wainwright, pp. 325–349, Boca Raton, FL: CRC Press. [1]

Kingma, D. P., and Ba, J. (2015), "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, eds. Y. Bengio and Y. LeCun. [8]

Kingma, D. P., and Welling, M. (2013), "Auto-Encoding Variational Bayes," arXiv preprint arXiv:1312.6114. [8]

Li, C., and Fan, X. (2019), "On Nonparametric Conditional Independence Tests for Continuous Variables," *Wiley Interdisciplinary Reviews: Computational Statistics*, 12, e1489. [1]

Li, C., Shen, X., and Pan, W. (2020), "Likelihood Ratio Tests for a Large Directed Acyclic Graph," *Journal of the American Statistical Association*, 115, 1304–1319. [1,3,10]

Liang, T. (2018), "On How Well Generative Adversarial Networks Learn Densities: Nonparametric and Parametric Results," arXiv preprint arXiv:1811.03179. [9]

McDonald, D. J., Shalizi, C. R., and Schervish, M. (2015), "Estimating Beta-Mixing Coefficients via Histograms," *Electronic Journal of Statistics*, 9, 2855–2883. [9]

Meinshausen, N., Meier, L., and Bühlmann, P. (2009), "P-values for High-Dimensional Regression," *Journal of the American Statistical Association*, 104, 1671–1681. [7]

Menon, V. (2011), "Large-Scale Brain Networks and Psychopathology: A Unifying Triple Network Model," *Trends in Cognitive Sciences*, 15, 483–506. [13]

Pearl, J. (2009), *Causality*. Models, Reasoning, and Inference (2nd ed.), Cambridge: Cambridge University Press. [9]

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011), "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, 2825–2830. [10]

Peters, J., Mooij, J. M., Janzing, D., and Schölkopf, B. (2014), "Causal Discovery with Continuous Additive Noise Models," *Journal of Machine Learning Research*, 15, 2009–2053. [2,3]

Power, J. D., Cohen, A. L., Nelson, S. M., Wig, G. S., Barnes, K. A., Church, J. A., Vogel, A. C., Laumann, T. O., Miezin, F. M., Schlaggar, B. L., and Petersen, S. E. (2011), "Functional Network Organization of the Human Brain," *Neuron*, 72, 665–678. [13]

Qiu, H., Han, F., Liu, H., and Caffo, B. (2016), "Joint Estimation of Multiple Graphical Models from High Dimensional Time Series," *Journal of the Royal Statistical Society*, Series B, 78, 487–504. [5]

Robins, J. M., Li, L., Mukherjee, R., Tchetgen, E. T., and van der Vaart, A. (2017), "Minimax Estimation of a Functional on a Structured High-Dimensional Model," *The Annals of Statistics*, 45, 1951–1987. [9]

Romano, J., and DiCiccio, C. (2019), "Multiple Data Splitting for Testing," Technical Report. [5,7]

Sachs, K., Perez, O., Peter, D., Lauffenburger, D. A., and Nolan, G. P. (2005), "Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data," *Science*, 308, 523–529. [1]

Schmidt-Hieber, J. (2017), "Nonparametric Regression Using Deep Neural Networks with Relu Activation Function," arXiv preprint arXiv:1708.06633. [9]

Shah, R. D., and Peters, J. (2020), "The Hardness of Conditional Independence Testing and the Generalised Covariance Measure," *The Annals of Statistics*, 48, 1514–1538. [1,4]

Shi, C., Xu, T., Bergsma, W., and Li, L. (2021), "Double Generative Adversarial Networks for Conditional Independence Testing," *The Journal of Machine Learning Research*, 22, 13029–13060. [1]

Siebert, W. M. (1986), *Circuits, Signals, and Systems*, Cambridge, MA: MIT Press. [6]

Smith, S. D., Fox, P. T., Miller, K., Glahn, D., Fox, P., Mackay, C. E., Filippini, N., Watkins, K. E., Toro, R., Laird, A., and Beckmann, C. F. (2009), "Correspondence of the Brain; Functional Architecture During Activation and Rest," *Proceedings of the National Academy of Sciences of the United States of America*, 106, :13040–13045. [13]

Spirtes, P., Glymour, C., and Scheines, R. (2000), *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning (2nd ed.), Cambridge, MA: MIT Press. With additional material by David Heckerman, Christopher Meek, Gregory F. Cooper and Thomas Richardson, A Bradford Book. [1,4]

Tsiatis, A. (2007), *Semiparametric Theory and Missing Data*, New York: Springer. [2]

van de Geer, S., and Bühlmann, P. (2013), "ℓ_0-penalized Maximum Likelihood for Sparse Directed Acyclic Graphs," *The Annals of Statistics*, 41, 536–567. [1]

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., and WU-Minn HCP Consortium. (2013), "The WU-Minn Human Connectome Project: An Overview," *Neuroimage*, 80, 62–79. [12]

Van Praag, C. D. G., Garfinkel, S. N., Sparasci, O., Mees, A., Philippides, A. O., Ware, M., Ottaviani, C., and Critchley, H. D. (2017), "Mind-Wandering and Alterations to Default Mode Network Connectivity When Listening to Naturalistic Versus Artificial Sounds," *Scientific Reports*, 7, 45273. [13]

Wang, Y., Kang, J., Kemmer, P. B., and Guo, Y. (2016), "An Efficient and Reliable Statistical Method for Estimating Functional Connectivity in Large Scale Brain Networks Using Partial Correlation," *Frontiers in Neuroscience*, 10, 1–17. [5]

Yu, Y., Chen, J., Gao, T., and Yu, M. (2019), "Dag-gnn: Dag Structure Learning with Graph Neural Networks," in *International Conference on Machine Learning*, pp. 7154–7163. [1,7]

Yuan, Y., Shen, X., Pan, W., and Wang, Z. (2019), "Constrained Likelihood for Reconstructing a Directed Acyclic Gaussian Graph," *Biometrika*, 106, 109–125. [1,3]

Zhang, H., Zhou, S., and Guan, J. (2018), "Measuring Conditional Independence by Independent Residuals: Theoretical Results and Application in Causal Discovery," in *Thirty-Second AAAI Conference on Artificial Intelligence*. [4]

Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. (2018), "Dags with No Tears: Continuous Optimization for Structure Learning," in *Advances in Neural Information Processing Systems*, pp. 9472–9483. [1,3]

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., and Xing, E. P. (2020), "Learning Sparse Nonparametric DAGs," in *International Conference on Artificial Intelligence and Statistics*. [1,3,5,7,10]

Zhu, S., Ng, I., and Chen, Z. (2020), "Causal Discovery with Reinforcement Learning," in *International Conference on Learning Representations*. [1,7,10]