Influencers, inefficiency and fraud - The Bitcoin price discovery network under the microscope *

Simon Trimborn †[1,2], Ying Chen[3,4], and Ray-Bing Chen[5]

[1]Amsterdam School of Economics, University of Amsterdam
[2]Department of Management Sciences, City University of Hong Kong
[3]Department of Mathematics, National University of Singapore
[4]Risk Management Institute, National University of Singapore
[5]Department of Statistics, National Cheng Kung University, Taiwan

December 8, 2022

Abstract

We present a TriSNAR modeling framework for understanding the dynamic interactions of multiple markets for Bitcoin trading, including market efficiency, and for identifying influential exchanges in the global trading network. We consider two types of influential exchanges from the perspectives of investors, regulators, and policymakers: exchanges that are market leaders and exchanges potentially used for market manipulation. Among 194 Bitcoin exchanges, we find that exchange Kraken was the leading exchange prior to the market frenzy of 2017. We also find a fraud-related exchange (Bitfinex) where some other exchanges display a similar role in the price discovery network than this exchange, raising questions about whether they may also be used for fraudulent activities. In addition, price discovery shows that the Bitcoin exchange network has been decreasing in efficiency from 2015 to 2017, and it has been increasingly efficient since 2018. We investigate the finite sample and asymptotic properties of TriSNAR. Compared to alternative methods, TriSNAR outperforms in terms of accuracy, runtime, and ability to discover multi-market network structures.

Keywords: Influencer Identification, Market Efficiency, Fraud Detection, Structure Detection, Bitcoin Exchanges

JEL classification: C01, C55, C58, G14

*The authors are thankful for helpful discussions and comments by Frank Kleibergen, Roland Füss, Geneviève Gauthier, Bas Werker, Otilia Boldea, and many others, including the seminar participants at City University of Hong Kong, Humboldt Universität zu Berlin, EmLyon, IESEG, University of Amsterdam, Tilburg University, University of St. Gallen, HEC Montreal and University of Waterloo.

†Corresponding author, phone: +31 643 611 771, E-Mail: trimborn.econometrics@gmail.com

1 Introduction

Technological advances and globalization have begun to reshape the trading and investment ecosystem. One notable change is the increase in cross-listing of financial products on multiple exchanges to raise more capital and liquidity. This market fragmentation, coupled with modern developments in exchange infrastructure, has been accompanied by billions of orders and executions. It has become increasingly important for investors, regulators and policy makers to assess and understand the impact of dynamic multi-market interactions and information flows in large-scale global financial networks. When each exchange is viewed as an individual in a global trading network, its price formation is not only subject to local supply and demand, but may be influenced more by price information from its competing markets. On some of these exchanges, the competing markets are considered to be signal providers for dynamic price discovery. They become more influential by leading future price movements and opening up arbitrage opportunities in the market, see (Eun and Shim, 1989; Lin et al., 1994; Asgharian et al., 2013; Chen et al., 2002). Meanwhile, monitoring of some exchanges is important because fraud on these exchanges, such as market manipulation, can have a strong spillover effect on global price discovery. This has prompted thinking about how to identify these influential exchanges and the role they play in price discovery and market efficiency.

An increasingly important financial multi-market trading network is the widely unregulated Bitcoin (BTC) market. BTC is traded simultaneously on a plethora of exchanges worldwide. These exchanges trade BTC against currencies such as the US dollar or other cryptocurrencies (CCs) including the stablecoin USDT, which is pegged to the US dollar. Unlike traditional stock exchanges, the BTC market has neither market closures nor national borders. The synchronized global trading of BTC with other currencies and the increasing importance of Bitcoin due to events like the recent emergence of being made official currency of El Salvador, has prompted thinking about which of the exchanges are more influential. As nodes in the network, the trading activity of these exchanges affects price discovery, influencing the price of the entire BTC trading network and further affecting market efficiency.

Extensive research has been conducted to study the dynamic behavior of networks consisting of multiple CC exchanges on global price discovery. Makarov and Schoar (2019) investigate a network of five CC exchanges during the 2017 market frenzy, from which they found one exchange (Bitfinex) to have a large impact on price discovery. Alexander et al. (2020) find that the BitMEX derivatives exchange has a leading effect on BTC spot prices on three exchanges. Makarov and Schoar (2020) examine arbitrage on CC exchanges. Capponi and Jia (2021) analyze decentralized exchanges and the implications of the use of automated market makers. Griffin and Shams (2020) find that market frenzy in 2017 was

accompanied by price manipulation of the BTC/USDT pricing series. Cong et al. (2021) examine the problem of fake liquidity on CC exchanges, also known as wash trading. These findings highlight the importance of understanding the role of multiple exchanges in BTC price discovery. In this study we are investigating a network of 194 CC exchanges which trade either BTC against USD or USDT, the latter being a stablecoin pegged to the USD, for their lead behaviour on the BTC price discovery. The results contribute to a better understanding of BTC price discovery, influential exchanges and market efficiency, as well as market price manipulation.

The study of the dynamic interactions of multi-market networks is not new. In the finance, economic and econometric literature, the dynamic interactions of various international networks have been investigated, such as economic exposure networks (Pesaran et al., 2004), EU networks (Dees et al., 2007), risk networks spanned by financial firms (Diebold and Yılmaz, 2014; Härdle et al., 2016), exchange rate and credit risk ratings (Creal et al., 2013), company networks (Rapach et al., 2013; Barigozzi and Brownlees, 2019), social networks (Chen et al., 2021; Kline and Tamer, 2017; Zhu et al., 2017), and Euro-zone bank networks (Bonaldi et al., 2015). The role of multi-market networks in price discovery has also been studied. Gagnon and Karolyi (2013) find that cross-listing in multiple markets matters for price discovery, while Chen et al. (2013) study the particular price discovery in two markets. Gagnon and Karolyi (2010) study the trading behavior of multiple markets and the resulting arbitrage opportunities. Halling et al. (2013) study the similarity of cross-listed firms in markets with high trading volumes. Lin et al. (1994), Asgharian et al. (2013), Chen et al. (2002), study the connectivity across sets of stock markets. In terms of cryptocurrency markets, Guo et al. (2019) study the network structure of cryptocurrency markets and Chen et al. (2018) consider the network connectivity within the BTC blockchain. While there are different underlying categories, the presence of dominant entities and interactions suggests that the answer to the question of multi-market price discovery mechanisms relies on the investigation of lead-lag time effects in the span of these networks. This has motivated the use of vector autoregressive (VAR) models to describe the network dynamics between multiple series and facilitate their interpretation, providing valuable information for inference and aiding prediction accuracy (Tsay, 2016).

However, for high dimensions or when multiple lags are involved, VAR models quickly become overparameterized and experience difficulties in convergence. Even for a moderate number of dimensions, the model structure is often overparameterized. This leads to potentially inaccurate estimates and also impairs the understanding and interpretation of the model. Basu and Michailidis (2015) argue that meaningful (interpretable) estimation and inference of large-scale VAR models is often not possible without imposing structural assumptions. The sparse structure of the parameters present in economic and social networks has been supported by a large body of empirical evidence and domain knowledge,

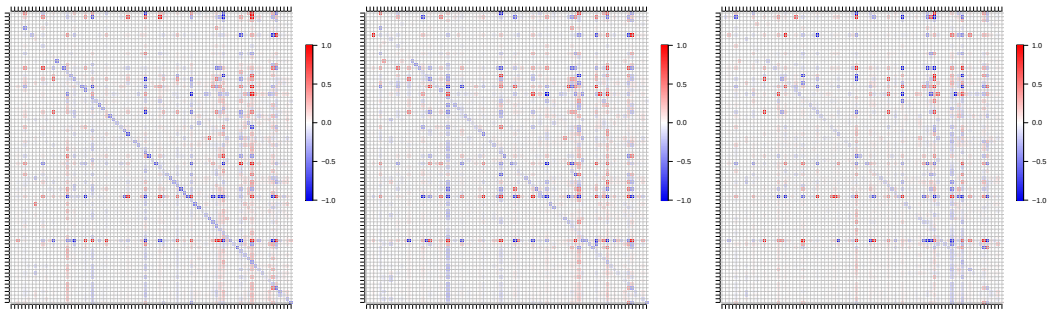
(de Paula, 2017). This also applies to networks where BTC is traded in multiple markets.

A plethora of estimators have been developed based on the assumption of sparse structure. Tibshirani (1996) proposes the least absolute shrinkage and selection operator (LASSO) for sparse estimation using a soft threshold operator. Other approaches suggest the use of hard threshold functions; see Antoniadis (1997). To overcome the discontinuity of the hard threshold operators, several unbiased and continuous regularization estimators have been proposed, such as the smoothed clipped absolute deviation (SCAD) (Fan and Li, 2001) and an adaptive LASSO (Zou, 2006). All these estimators are proposed for regression models. For multivariate time series data, Song and Bickel (2011) propose a two-stage VAR approach that penalizes the lagged parameter matrix, columns, and individual parameters, respectively. Davis et al. (2016) develop a two-step approach for network autoregressive (NAR) models to study dynamic interconnections of large-scale networks under sparsity. Both methods translate into a hard threshold operator. Nicholson et al. (2017) consider various structured VAR models with LASSO and sparse group penalty functions (Simon et al., 2013) to construct soft threshold estimators. Basu et al. (2019) propose a VAR(1) model for reducing the rank and parameter of the underlying structure such as the network structure matrix. Lin and Michailidis (2017) study block VAR models, while Skripnikov and Michailidis (2019) combine a group LASSO and LASSOs to jointly estimate several VAR models. Bayesian VAR models have been extensively studied; see, for example, Ghosh et al. (2019).

Figure 1 shows the estimated parameter matrices for BTC trading on the 80 most liquid exchanges by average trading volume from January 1, 2016 to December 31, 2018 under the VAR (3) framework. We see that most of the columns contained in each matrix are shown in light red/blue, which indicates that the respective parameters are very close to zero, i.e., there is no connectivity. At the same time, some columns have many parameters that differ significantly from 0, reflecting the emergence of sparsity in the BTC multi-market trading network structure, i.e., one exchange has an impact on some exchanges but may not have an impact on others. Furthermore, the serial dependence between price discovery at different times raises the question of whether all the three lags are needed, or whether fewer lags are sufficient to describe the underlying lead-lag links. The location, type and structure of these parameters are usually unknown. This motivates us to define a network autoregressive (NAR) model that applies regularized estimation to the network of transactions between exchanges. It contains three types of sparsity in the temporal network:

- lags that model the temporal impact of the entire multi-market network on all exchanges,
- columns in the active NAR parameter matrix, reflecting the group effect of one exchange on all other exchanges (influencers), and

Figure 1: The 3 matrices are the VAR(3) parameter matrices of the top 80 exchanges considered in the study by their trading volume, estimated on the interval 01.01.2016 until 31.12.2018. The left matrix corresponds to lag 1, the right one to lag 3.



- elements within the activity group, reflecting individual effects, i.e. interactions between any two exchanges.

To study price discovery and market efficiency among BTC exchanges, we propose a model specifically designed to identify leading exchanges (influencers) in multi-market networks, called the 3- layer Sparse estimator for large-scale Network AutoRegressive models (TriSNAR). In contrast to classical VAR model, TriSNAR aims to detect the network structures in order to decipher their economic implications, e.g., to find influential nodes in a dynamic network structure. The estimation is performed under triple sparsity of lags, groups and individuals. We develop efficient and fast algorithms that sequentially optimize the large-scale estimation for each of the three layers. We also provide an approximate optimization algorithm to achieve fast convergence of the regularized estimates. We determine asymptotic properties of the sparse estimator under the assumption of fixed dimensionality and show that TriSNAR has the three properties of a good sparse estimator as defined by Fan and Li (2001), namely, unbiasedness, sparsity, and continuity. In an extensive experiment with synthetic data, TriSNAR provides good accuracy while revealing the true structure, which greatly improves the interpretability of the dynamic structure of the network.

There are several existing sparse estimators for VAR/NAR models. TriSNAR differs from them in the following aspects. Davis et al. (2016) and Kock and Callot (2015) focus on the individual effects between the two processes. The sparse estimator neither cover the group effect of one process on the entire network nor the serial dependence effect of the system. Chen et al. (2018) and Nicholson et al. (2017) consider any two of the three effects, but do not cover all of them. Song and Bickel (2011) promote triple sparsity; however, their approach does not possess the three properties of a good estimator.

We apply TriSNAR to price discovery in a BTC multi-market trading network. We find that many influential exchanges existed before the market frenzy in 2017, a vanishing

structure is observed in 2018, and from 2019 onwards, there are hardly any exchange leaders left to be identified. It is worth noting that exchange Kraken had a full year of influence on price discovery across the network, from April 2016 until March 2017. As such, it reveals the possible existence of market inefficiency. Based on the relationship between price discovery and market efficiency, we find that prior to 2018, the BTC market was inefficient provided various exchanges influenced prices. From 2018 onwards, the market efficiency of BTC has improved given the absence of influential exchanges.

We investigate market manipulation and fraudulent behaviours. Griffin and Shams (2020) argue that market prices were manipulated through the Bitfinex exchange during the 2017 market frenzy. Through TriSNAR, we determined that Bitfinex was the only influential exchange during this period, supporting the claim of price manipulation. In further investigation, we find that the exchange BitZ was influential in price discovery in Q4 2018, which raises questions about whether this exchange was also used to manipulate the market. Meanwhile, while intense market volatility also occurred in early 2021, our analysis shows that no exchange influenced price discovery during this period, including this particular exchange. It projects that the market frenzy in 2021 was structurally different from that of 2017.

This paper is structured as follows: In Section 2 we describe and analyse the data. We present TriSNAR, the model framework and provide a detailed description of the penalization operator in Section 3. In Section 4, we study the BTC price discovery network with TriSNAR and various other network models, in regards to influential exchange identification, price discovery network efficiency and fraud detection. In Section 5, we investigate the finite-sample performance in a synthetic data study. Section 6 concludes. The code used in this paper is available at GitHub.

2 Data description

In this study we consider the network relations between 194 BTC exchanges over a time period of 5.5 years. Among the exchanges, not all allow trading against the USD, e.g., Poloniex and Binance, some only offer trading against other CCs including stablecoins like USDT which is pegged to the USD. To study the behaviour of the price dynamics between the exchanges and to identify influential exchanges which contribute to the price discovery, we collect hourly Bitcoin time series in USD and USDT from 194 CC exchanges in the time period 01.01.2015 until 11.05.2021. Since USDT is pegged to the USD, the BTC/USDT time series can be treated as similar to BTC/USD. Some exchanges offer trading of Bitcoin against both, USD and USDT. In this situation we take the weighted average of the two time series, whereas we weight the two time series by their trading volume occurring at the

particular hours. By this we have a single exchange rate BTC/USD for each of the 194 exchanges.

As we are interested in studying the price discovery between the exchanges, we start by analysing the exchanges for their time series properties. Various of the exchanges are only active for a fraction of the overall time period, which is a challenge for a multivariate time series analysis. To overcome this difficulty, we intend to split the dataset into subsets and analyse the network structure within the subsets. Further various of the exchanges are subject to low trading volume. For a first investigation, we fit a VAR(3) model to the returns of the 80 most liquid exchanges over two different time periods, namely 01.01.2016 until 31.12.2018 and 01.01.2019 until the end of the time period. We cut off 2015 because only a few exchanges are active in that year and their inclusion would lower the dimension of the VAR model too much. We plot the parameters of the VAR(3) as a heatmap and for illustration purposes we constrained parameters outside of the interval $[-1, 1]$ to the interval boundaries, see Figure 2. A column in each heatmap contains the parameters at $t - k$ with $k \in \{1, 2, 3\}$ of one exchange as related to the returns at t of all other exchanges. A column with parameters almost entirely different from 0 indicates that exchange is leading the price discovery on the other exchanges, hence the exchange acts as an influencer. Vice versa if a row has time lagged influence of all exchanges towards a single one, it indicates that one exchange lags behind the others in the price evolution. We observe that only a few columns have parameters which are clearly different from 0 which indicates the presence of leading exchanges. Further we observe a difference in the parameter matrices between the two time periods which suggests that the relationship is not constant over time, instead dynamic in nature. For various columns the parameters are close to 0, which indicates that a sparse estimation of the matrices is called for.

In the data matrix the exchanges are ordered by the size of the average trading volume. Since the exchanges with a larger influence are scattered over the full spectrum of the parameter matrix, this indicates that trading volume is not a predictor for a leadership role of an exchange in the price discovery network. In Figure 3 we compare the average trading volume on an exchange against the standard deviation in its trading volume. We observe that a high trading volume usually also comes with a higher variation and that there are some exchanges with considerably higher trading volume than others. The sparsity in the VAR parameter matrix estimation and the missing link between a high trading volume and a leadership role in the price discovery network motivates an investigation of the network with sparse network methods.

We established that the a lead-lag relationship exists between the exchanges which is dynamic over time. Taking into account that investors observe the price evolution on all the exchanges and incorporate the corresponding information into their trading decisions, we address the following questions:

Figure 2: Upper 3 matrices are the VAR(3) parameter matrices estimated on the interval 01.01.2016 until 31.12.2018, lower 3 matrices are estimated on the interval 01.01.2019 until 11.05.2021. The left matrix corresponds to lag 1, the right one to lag 3.

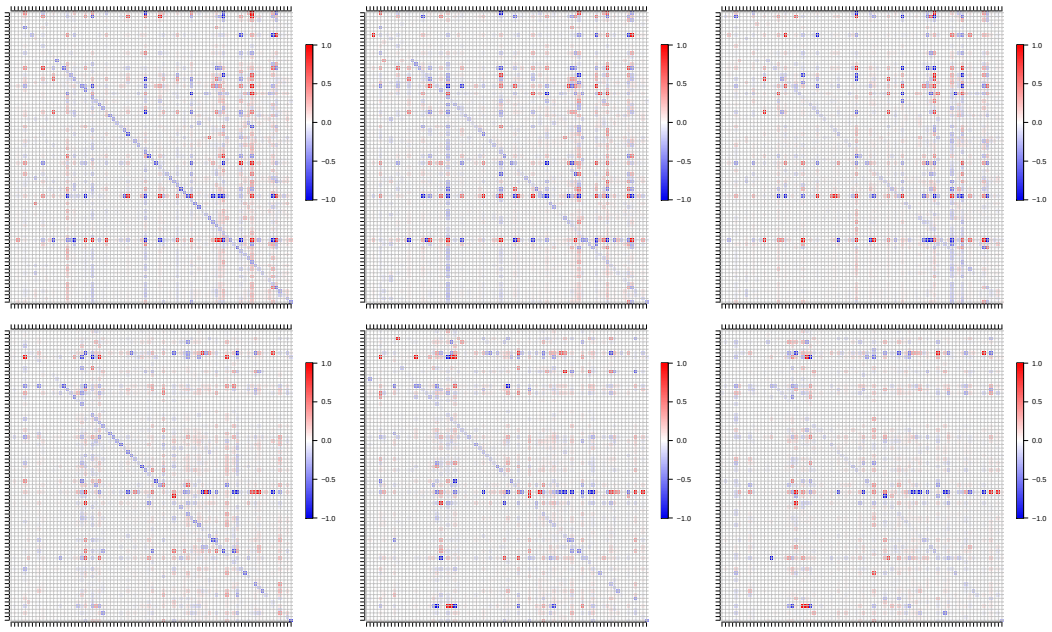
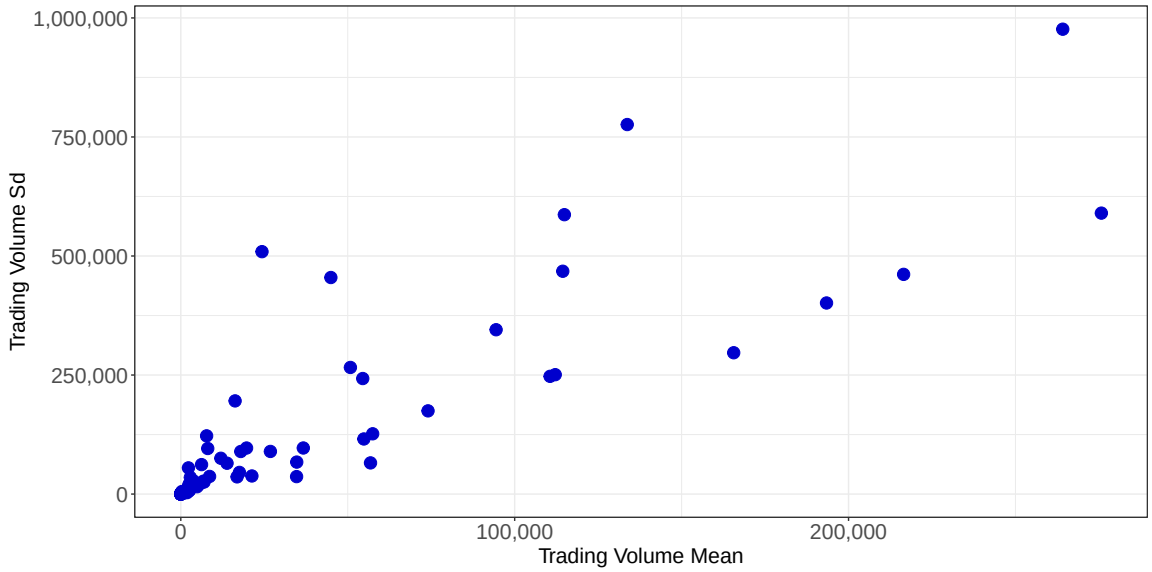


Figure 3: The average trading volume on 80 exchanges plotted against its standard deviation in the trading volume in the time period 01.01.2016 until 11.05.2021.



- The inter-exchange network between traders on different exchanges is not directly observable, although respective traders observe the price evolution on other exchanges. This raises the question of whether there is information flow such that traders on certain exchanges influence those on others on a regular basis, and thus, certain exchanges essentially serve as influential network nodes?
- Is time a factor, or, more specifically, do multi-market interactions last for only one period or are they continuously present over several periods?
- What is the relationship between arbitrage and the leadership of exchanges? Do different exchanges contribute to the price discovery when arbitrage opportunities widen/narrow?
- Is a change in the network relation between exchanges connected to the overall market liquidity?
- When fraudulent behaviour appears on an exchange, that exchange would take on a leading role in the price discovery network since the price manipulation would make it the dominating network node. Hence, are the identified leading exchanges in the network linked to price manipulation?
- How is the connectedness of exchanges related to the state of the CC market?

We established that the data are subject to sparsity with many parameters close to 0. Motivated by this and to study the price discovery between the Bitcoin exchanges, we are going to apply the LASSO and SCAD methods to study the relationship between the exchanges. However these methods do not take into account that the different lags of a VAR model have an inherent different relation to the contemporaneous observations, which leads us to use the 'tapered LASSO' method of Nicholson et al. (2017) as well. Though none of these methods are designed to uncover influencers in the network structure. As observed in Figure 2, certain exchanges appear to act as influencers and hence have leadership roles in the price discovery network. As to uncover such network structures, we are proposing a network estimator called TriSNAR, Three-Layer Sparse Network AutoRegression. The estimator, which we are introducing in Section 3, is designed to uncover leading nodes (influencers) in the described network structures and we are going to study its theoretical properties as well as show its finite sample performance as compared to the three other methods used in this study.

3 TriSNAR for Large-Scale Network AutoRegressive Models

Let $Y_t \in \mathcal{R}^d$ denote a vector of observations of a network with d-dimensional processes at time point $t \in \{1, \ldots, T\}$, with the length of the time period T. Assume there are p

parameter matrices $\boldsymbol{A}_k$ with $k \in \{1, \ldots, p\}$, which are of dimension $d \times d$ and measure the serial dependence of the d processes. The Network AutoRegressive (NAR) model describes the serially linear dependencies between these processes. The matrices $\boldsymbol{A}_k$ are assumed to have a sparse 'network' structure, as illustrated in Sections 1 and 2, and thus some variables have no influence on others at all. This means, the parameters are $A_{k;ij} = 0$ for some k, j and all i. Given time lag k, $\boldsymbol{A}_k$ has either of the following structures:

$$\boldsymbol{A}_k \begin{cases} A_{k;ij} = 0, & \forall i, j \\ A_{k;ij} = 0, & \forall i \text{ and some } j \wedge A_{k;ij} \neq 0 \text{ otherwise} \\ A_{k;ij} = 0, & \text{for some } i, j \ \wedge A_{k;ij} \neq 0 \text{ otherwise} \\ A_{k;ij} \neq 0 & \text{otherwise} \end{cases} \tag{1}$$

where the first two cases inspire the need of a NAR model. In the first case, there is no influence from any variable j to any i (no influential network effect in lag k). In the second case, some variables j do not influence any i (no influential network effect from variable j at lag k). In other words, some variables do not affect others at all or just at certain time lags, which represents no-influential-network effect. Others are influential network variables (influencers) and the influence of these variables j on variables i at time lag k will be $A_{k;ij} \neq 0$. This definition relates to Figure 1 and the sparse network structure in the corresponding VAR model. The NAR is defined as

$$Y_t = A_0 + \sum_{k=1}^{p} \boldsymbol{A}_k Y_{t-k} + \epsilon_t \tag{2}$$

where $A_0 = (a_{1,0}, \ldots, a_{d,0})^\top$ is the intercept and $\epsilon_t = (\epsilon_{1,t}, \cdots, \epsilon_{d,t})^\top$ is a vector that is assumed to be independently and identically distributed with $\epsilon_t \sim (0, \Sigma)$. We assume that the model is stationary and ergodic, with all roots of the polynomial $\boldsymbol{I}_d - \sum_{k=1}^{p} \boldsymbol{A}_k Z^k$ lying outside the unit ball. Though the definition is in a VAR framework, it differs in the described influencer network structure of the parameter matrices, see equation (1).

Our interest is in detecting the influential variables and their dynamic dependence structure in the network to help us understand the multi-market interactions. For an insightful interpretation, we assume a sparse network structure, which is motivated by Figure 2. In other words, the parameter matrices $\boldsymbol{A}_k$ for $k = \{1, \ldots, p\}$ are sparse, whereas the location and form of the sparsity are not predetermined. To perform regularized estimation, we introduce a penalty function $p_{\lambda_1,\lambda_2,\lambda_3}(\cdot)$ imposed on the lags, columns and individual parameters of $\boldsymbol{A}_k$ which is designed to detect structures as described in equation (1). We

estimate the model (2) by solving a regularized least squares optimization problem,

$$\min_{\mathbf{A}} \sum_{t=p+1}^{T} \frac{1}{2}||Y_t - \sum_{k=1}^{p} \boldsymbol{A}_k Y_{t-k}||_F^2 + \sum_{k=1}^{p} p_{\lambda_1,\lambda_2,\lambda_3}(\boldsymbol{A}_k), \tag{3}$$

where λ_1, λ_2 and λ_3 are tuning parameters for the sparsity and $||\cdot||_F$ is the Frobenius norm, defined by $||\boldsymbol{A}_k||_F = \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d} A_{k;ij}^2}$, with $A_{k;ij}$ the i,jth entry of the matrix $\boldsymbol{A}_k$. The penalty function, for a suitable choice of λs, should permit the estimator to have the properties of unbiasedness, sparsity and continuity and, further, the oracle property.

We extract the diagonal of $\boldsymbol{A}_k$ and consider the autoregressive parameters separately as the $(d+1)$th group, $A_{d+1;j}$. In other words, we describe the autoregressive effects disentangled from the network effects. We use $A_{k;j}$ to denote the column (group) j with $j = 1, \ldots, d$ within the parameter matrix $\boldsymbol{A}_k$, yet without the jth parameter on the diagonal. Hence, $A_{k;j} = (A_{k;1j}, \ldots, A_{k;(j-1)j}, A_{k;(j+1)j}, \ldots, A_{k;dj})^\top$. As such, the groups have $(d-1)$ parameters except for the group of the autoregressive parameters, which has d parameters. We introduce a scaling parameter $d_j = (d-1)$ for $j = 1, \ldots, d$ and $d_j = d$ for $j = (d+1)$ to offset the impact of a mismatch between the number of parameters in the columns $(d-1)$ and the diagonal d and apply it to the regularization parameter λ_2 accordingly. We define the penalty function by

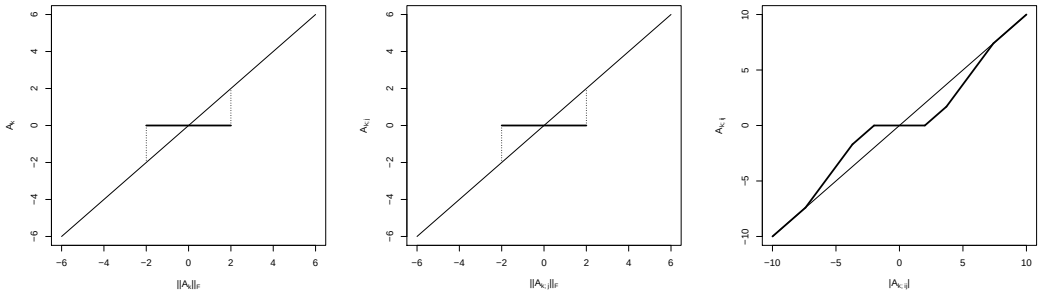
$$p_{\lambda_1,\lambda_2,\lambda_3}(\boldsymbol{A}_k) = \begin{cases} d^2\lambda_1||\boldsymbol{A}_k||_F & ||\boldsymbol{A}_k||_F \leq d^2\lambda_1 \\ d_j\lambda_2||A_{k;j}||_F & ||A_{k;j}||_F \leq d_j\lambda_2 \wedge d^2\lambda_1 < ||\boldsymbol{A}_k||_F \\ \lambda_3|A_{k;ij}| & |A_{k;ij}| \leq \lambda_3 \wedge d_j\lambda_2 < ||A_{k;j}||_F \wedge \\ & d^2\lambda_1 < ||\boldsymbol{A}_k||_F \\ \frac{2b\lambda_3|A_{k;ij}| - |A_{k;ij}|^2 - \lambda_3^2}{2(b-1)} & \lambda_3 < |A_{k;ij}| \leq b\lambda_3 \wedge d_j\lambda_2 < ||A_{k;j}||_F \wedge \\ & d^2\lambda_1 < ||\boldsymbol{A}_k||_F \\ \frac{\lambda_3^2(b+1)}{2} & b\lambda_3 < |A_{k;ij}| \wedge d_j\lambda_2 < ||A_{k;j}||_F \wedge \\ & d^2\lambda_1 < ||\boldsymbol{A}_k||_F \end{cases}. \tag{4}$$

Note that the first case $d^2\lambda_1||\boldsymbol{A}_k||_F$ applies to the layer of lags. It is regularized by the magnitude of all parameters within $\boldsymbol{A}_k$ and scales the regularization parameter λ_1 by the number of parameters, d^2. The second case, $d_j\lambda_2||A_{k;j}||_F$, regularizes each group j of $\boldsymbol{A}_k$. The 3rd to 5th cases build the regularization operator for the individual parameters first, the soft- and tapering-off threshold, $\lambda_3|A_{k;ij}|$ and $\frac{2b\lambda_3|A_{k;ij}| - |A_{k;ij}|^2 - \lambda_3^2}{2(b-1)}$, and the non-regularized case $\frac{\lambda_3^2(b+1)}{2}$. Here, we require $b > 2$.

The penalty function (4) combines the advantages of two hard-thresholding functions

Figure 4: The three plots illustrate the three states of TriSNAR. The first plot shows the penalization of each k lag $\boldsymbol{A}_k$, which is penalized with a hard-thresholding function. When the penalizing value λ_1 is reached, the values are unpenalized. The second penalization function for the group steps is illustrated in the second figure, which is also a hard-thresholding function. Next, the individual penalization function becomes active, which corresponds to the form of SCAD for regression models.

(a) Lag penalization (b) Group penalization (c) Individual penalization



and a soft-thresholding and a tapering-off function. Figure 4 shows the performance of the penalization for examples of sequences of λ by setting the scaled tuning parameters to 2 and $b = 3.7$. The hard-thresholding functions are applied to the lagged parameter matrices (Fig. 4a) and the groups (Fig. 4b). The soft-thresholding and tapering-off functions, similar to the SCAD penalty for regression models, are used for the individual parameters (Fig. 4c). This function ensures that $\boldsymbol{A}_k$ and $A_{k;j}$ are only penalized until the values $d^2\lambda_1$ and $d_j\lambda_2$ are reached. This favors the unbiasedness of the resulting NAR model. This formulation is also grounded in the rationale that a group inside of a lagged matrix can only be unpenalized when the entire matrix is not subject to the hard-thresholding parameter λ_1.

Applying the three-layer penalty function (4), we obtain the estimator of $\boldsymbol{A}_k$ according to several cases:

$$\boldsymbol{A}_k = \begin{cases} 0 & ||\boldsymbol{A}_k||_F \leq d^2\lambda_1 \\ 0 & ||A_{k;j}||_F \leq d_j\lambda_2 \ \wedge \ d^2\lambda_1 < ||\boldsymbol{A}_k||_F \\ sgn(A_{k;ij})(|A_{k;ij}|-\lambda_3)_+ & |A_{k;ij}| \leq 2\lambda_3 \ \wedge \ d_j\lambda_2 < ||A_{k;j}||_F \wedge d^2\lambda_1 < ||\boldsymbol{A}_k||_F \\ \frac{(b-1)A_{k;ij}-sgn(A_{k;ij})b\lambda_3}{(b-2)} & 2\lambda_3 < |A_{k;ij}| \leq b\lambda_3 \ \wedge \ d_j\lambda_2 < ||A_{k;j}||_F \wedge d^2\lambda_1 < ||\boldsymbol{A}_k||_F \\ A_{k;ij} & b\lambda_3 < |A_{k;ij}| \wedge d_j\lambda_2 < ||A_{k;j}||_F \wedge d^2\lambda_1 < ||\boldsymbol{A}_k||_F \end{cases} \tag{5}$$

The estimator is designed to detect influential variables in the network, hence to detect a structure as defined in equation 1. We propose two algorithms for the estimation of

TriSNAR, one being an active-set optimization algorithm denoted by TriSNAR$_G$ and an approximate one denoted by TriSNAR$_A$. We introduce TriSNAR$_A$ due to its improved runtime while it usually provides a similar result than TriSNAR$_G$ as we show in the synthetic data study, see section 5.

4 Bitcoin Price Discovery Network Analysis

We are going to analyse the network structure between the exchanges for their lead-lag behaviour with a focus upon influential exchanges over various time periods and relate the results to price discovery and market manipulation. Our results reveal a higher inefficiency in the market before the December 2017 market frenzy, since we found exchanges acting as market leaders. The exchange Kraken stands out since it is identified for an entire year as influential for the price discovery network. We relate the discovery of influential exchanges to the availability of arbitrage opportunities in the market at these times. We were able to identify a single market leader while the December 2017 market turbulences, namely Bitfinex, which is known for being the platform for BTC/USDT exchange rate manipulations during that time period, Griffin and Shams (2020). In various other time periods, we observe a similar network structure as the one observed for Bitfinex. For example in 2018, we observe a similar outstanding role of the exchange BitZ, which gives rise to the question if fraudulent behaviour took place on that exchange as well. From 2019 onwards, we observe a higher efficiency in the market with hardly any exchanges influencing the market. Notably, before 2018 usually TriSNAR provided the best model in terms of BIC compared to the other model contestants though from 2018 onwards mostly a different model provided the best BIC. Since TriSNAR focuses on the identification of influencers in networks, this underscores the observation that the influencer network structure vanished after 2018. We conclude that the markets price discovery network became first increasingly inefficient until the 2017 market frenzy, after which it became increasingly efficient. Overall the network analysis and TriSNAR in particular aided us to identify influential exchanges, identify exchanges which are linked to market manipulation and allowed us to conclude that the market improved its efficiency over time.

4.1 Set-up

We employ TriSNAR to study the dynamic connections within the exchange network by splitting the data into training and validation sets. In the study we use the proposed algorithm TriSNAR$_G$, see Appendix I for the details of the algorithm. We split each year into quarters and we take the first two months of each quarter for the model estimation and the last month for model validation, which means we evaluate the regularization parameters

λ_1, λ_2, λ_3 on the latter dataset. E.g., for the first quarter of the year 2015, we would estimate the models on the time period 01.01.2015 until 28.02.2015 and validate the models on 01.03.2015 until 31.03.2015. The λ_1, λ_2, λ_3-sequences are generated from deriving the values first for which all parameters are set to 0. From this value a halving sequence is created for each λ until 0 is reached. For the analysis, we are deriving the log-returns from the price series. To make an easy and interpretable comparison of the estimated parameters possible, we adopt a GARCH(1,1) model to scale the demeaned data to unit variance, hence the magnitudes of the parameters become comparable between exchanges and over time. After data preparation, we perform an ADF and PP (Phillip-Perron) test on each training dataset, see Said and Dickey (1984) and Nelson and Plosser (1982). For each dataset the H_0 of non-stationarity gets rejected. After the model estimation, we test the residuals for stationarity with the same two tests. Again, each time the H_0 of non-stationarity gets rejected.

The dataset spans two time periods of particular interest, which are the market frenzies of November/December 2017 and December 2020/January 2021. As to analyse the network structure between the exchanges in this time periods, we subset the data so that the model evaluation takes place while the market is in the wild stage until it peaked, and the estimation of the models is performed on the 6 months of data before the date of the market frenzy. As for the first subset, we subset the data between the 24.05.2017 and 17.12.2017, and the second one between 11.06.2020 and 09.01.2021. For the first subset, we estimate the model on the time period 24.05.2017 until 23.11.2017 and validate them on 24.11.2017 until 17.12.2017. For the second subset, we estimate the model on 11.06.2020 until 10.12.2020 and evaluate them on 11.12.2020 until 09.01.2021. As before, we consider the log-returns and demean them as well as scale them with a GARCH(1,1) to unit variance. Since we intend to identify the true model, we evaluate the best fitting model on the evaluation dataset via the BIC criterion.

There exists a huge difference in the trading volume between the exchanges, see Figure 3. Exchanges with low trading volume cannot act as influencers for the price discovery, hence we focus in each time period on the exchanges which have at least an average trading volume of 5% of the most liquid exchange in that particular period.

4.2 Results & Chorddiagram

The results of applying TriSNAR, TLASSO, SCAD and LASSO on the time periods as described above are summarized in Tables 1 and 2. For the 4 methods, we report the absolute number of exchanges which were identified by the respective method as influential for the network as well as the influential exchanges share of the absolute number of exchanges considered in that time period. In accordance with our definition that influential exchanges

impact all or at least a significant number of other exchanges, we mark an exchange as influential if at least 25% of parameters associated with that exchange are nonzero. In Table 2 we report the BIC of the models and the number of selected parameters. We observe that the networks grow over time and that in general the share of influential exchanges from the total number decreases. We compare these results against the average trading volume between the exchanges in that particular time period. Similarly we also report the mean over the variance in the individual exchanges BTC/USD return series. Further we analyse the arbitrage by deriving the spread between the exchanges price series. The spread is derived from the mean price on each exchange, from which the difference between the min/max of the average prices is derived and divided by the average price between the exchanges. Provided that TriSNAR is especially designed to identify influential exchanges, we compare via BIC in Table 2 which model fits the data best in each quarter. In case no influential exchanges are present, either TLASSO, SCAD or LASSO are expected to outperform.

Several parameter matrices are shown to illustrate the results. The matrices are displayed as chorddiagrams. A chord diagram displays the direction and magnitude of the influence of each node (Bitcoin exchange) by showing the magnitude by means of the circle and the destination of the signal by the chord. The wider the space in the circle, the larger the magnitude and hence the higher the dynamic impact on the network. A chord diagram does not differentiate between positive and negative influences. The sum of the absolute values of the parameters (magnitude) is displayed on the circle.

4.3 2015 - 2017: Market inefficiency, arbitrage and influencers

We observe from Table 1 that the spread in the price between the exchanges increases from 2015 until Q2 of 2018. This opens up opportunities for arbitrage and would make it useful for investors to observe the price on various exchanges and act accordingly, hence the presence of influential exchanges could be expected. Indeed we observe that TriSNAR identified leading exchanges from 2015-2017. Notably often just one exchange was identified. For one year the exchange Kraken dominated the price discovery network, namely from Q2 2016 until Q1 2017, compare Figure 5. It is remarkable that this structure was observed for an entire year, hence the dominance is persistent. Before 2016, mostly LakeBTC took on leading roles in the price discovery network. The influence of these exchanges on the price discovery network is underscored by the other three models, TLASSO, SCAD and LASSO. In each of the models these exchanges strongly influence the network though the methods also assigned parameter weight to other exchanges. Table 2 shows that TriSNAR achieved the better BIC in this period in all but three quarters, showing that the detected influential exchanges from TriSNAR outperform the structure identified by the other three methods.

The presence and persistence of existence of influential exchanges relates to market

Table 1: Summary statistics of the prices on the exchanges, trading volume of the exchanges, and the absolute number and percentage of leading exchanges (minimum of 25% parameters different from 0) discovered via the network models. Spread refers to the difference between min/max in a quarter divided by average price in that quarter.

		Exchanges			Leading Exchanges							
		Price		Trading Volume	TriSNAR		TLASSO		SCAD		LASSO	
		Spread	Var	Average	Abs	%	Abs	%	Abs	%	Abs	%
2015	Q1	0.0077	0.0112	272825	2	100	2	100	2	100	2	100
	Q2	0.0055	0.0018	190238	3	50	5	83	4	67	6	100
	Q3	0.0158	0.0031	143881	1	20	3	60	2	40	4	80
	Q4	0.0165	0.0076	512852	2	33	6	100	6	100	2	33
2016	Q1	0.0142	0.0022	245754	3	33	5	56	1	11	1	11
	Q2	0.0212	0.0041	395426	2	25	5	62	2	25	2	25
	Q3	0.0222	0.0023	137468	1	10	3	30	1	10	2	20
	Q4	0.0189	0.0014	204339	1	9	5	45	1	9	1	9
2017	Q1	0.0163	0.0081	614978	1	11	3	33	3	33	3	33
	Q2	0.0407	0.0109	1772345	1	12	3	38	1	12	5	62
	Q3	0.0476	0.0122	3907513	6	75	6	75	2	25	5	62
	Q4	0.0595	0.0203	8419816	1	7	0	0	3	21	2	14
2018	Q1	0.0529	0.0234	13319677	0	0	0	0	10	83	0	0
	Q2	0.0514	0.3461	6663316	0	0	10	77	2	15	5	38
	Q3	0.0035	0.0296	6386750	0	0	0	0	0	0	0	0
	Q4	0.0362	0.0079	5365156	1	5	0	0	2	10	1	5
2019	Q1	0.025	0.0021	7629719	0	0	8	50	0	0	0	0
	Q2	0.0104	0.0093	19062786	1	5	0	0	1	5	4	18
	Q3	0.0026	0.0081	14143778	3	10	30	100	30	100	30	100
	Q4	0.0133	0.0089	14984220	0	0	0	0	0	0	0	0
2020	Q1	0.0036	0.0164	38316114	0	0	1	4	0	0	0	0
	Q2	0.0098	0.0049	28236537	0	0	0	0	0	0	0	0
	Q3	0.0014	0.0023	14567112	0	0	0	0	0	0	0	0
	Q4	0.0022	0.0044	27947155	2	9	0	0	2	9	2	9
	Q1	0.076	0.0132	79204079	0	0	0	0	0	0	0	0
	subset 2017	0.0252	0.0171	4697451	1	9	0	0	2	18	0	0
	subset 2021	0.0019	0.006	35701839	0	0	0	0	0	0	0	0

inefficiency. In particular that Kraken impacted the price discovery network for an entire year is a structural inefficiency which means that the price on that exchange was leading the network. It also raises the question why this particular exchange was influential. During this period Kraken was considered as a more trustworthy exchange than others in a market in which scamming of users is common. Also Kraken is located in the USA, whereas back then many other large BTC exchanges were listed in less regulated jurisdictions. These reasons could explain why this particular exchange was influential and the source of the market inefficiency.

We also observe that in this period the spread of prices between the exchanges increased, which means arbitrage opportunities were present. It is remarkable that the increase in spread was accompanied by a strong increase in trading volume as well. Leading exchanges were present in 2015 already, when the price spread was still relatively low. The price spread widened in 2016 & 2017, when Kraken was frequently the only influential exchange. However a wide price spread was also present in Q1 & Q2 of 2018, when no influential exchanges were detected. For the period Q3 2018 to Q1 2021, the price spread was comparably narrow and hardly ever influential exchanges were detected. Also, in that period mostly TLASSO

Table 2: BIC and percentage of number of parameters in model of the models TriSNAR (1), TLASSO (2), SCAD (3) and LASSO (4) in the respective quarters and subsets.

		BIC				percent numb para			
		1	2	3	4	1	2	3	4
2015	Q1	-1892	-1888	-1880	-1870	58	67	83	100
	Q2	-10804	-10785	-10605	-10487	19	29	26	44
	Q3	-2645	-2609	-2605	-2547	12	19	23	36
	Q4	-5804	-5653	-5637	-5557	11	24	31	7
2016	Q1	-7867	-7618	-7633	-7468	12	14	4	3
	Q2	-9633	-9757	-9469	-9442	19	27	11	20
	Q3	-6397	-6245	-6366	-6156	6	14	11	17
	Q4	-6196	-6029	-6087	-5914	13	16	16	9
2017	Q1	-10753	-10698	-10477	-10525	7	15	15	16
	Q2	-12588	-12834	-12567	-12611	6	16	7	20
	Q3	-12291	-12274	-12251	-12152	39	31	7	17
	Q4	-22189	-22226	-21627	-21985	2	2	7	6
2018	Q1	-30813	-30990	-30434	-30806	0	0	19	0
	Q2	-52627	-53861	-51580	-52635	2	35	11	15
	Q3	-47645	-47752	-47380	-47645	0	0	0	0
	Q4	-58780	-59367	-58497	-58482	3	1	3	2
2019	Q1	-36878	-37253	-36872	-36846	1	15	1	1
	Q2	-101446	-102351	-98002	-101073	2	1	3	5
	Q3	-530418	-517773	-513307	-513307	3	100	100	100
	Q4	-77103	-77795	-76886	-77062	0	0	1	0
2020	Q1	-64891	-64536	-63017	-64752	0	2	2	1
	Q2	-173651	-174538	-173422	-173594	0	0	1	1
	Q3	-96906	-97756	-96150	-96681	1	1	2	2
	Q4	-89468	-90523	-88688	-89108	2	0	4	4
	Q1	-76055	-76821	-76031	-76050	0	1	0	0
	subset 2017	-9579	-9489	-9414	-9435	6	1	3	2
	subset 2021	-160254	-160217	-160338	-159994	0	3	1	2

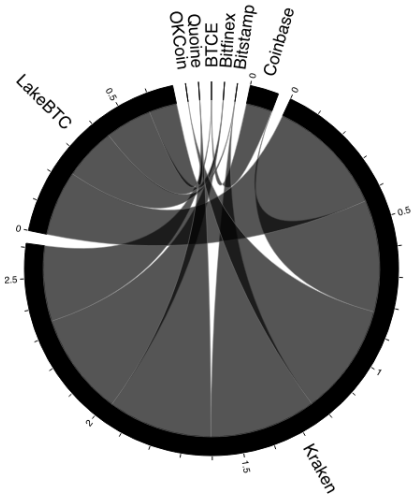
was found to provide the best models, see Table 2, which underscores the observation that the network structure vanished after 2017. As a result, it appears that influential exchanges are detected before arbitrage opportunities open up and vice versa no longer influential exchanges are detected before arbitrage opportunities vanish.

4.4 End 2017 - 2018: Market turbulences and fraud

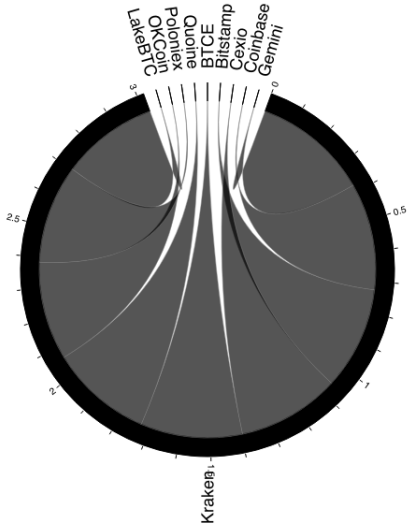
In the end of 2017, the CC market rose to new heights and the price of BTC experienced strong swings. Table 1 also shows that in Q4 2017 the spread between the exchanges was high but TriSNAR only identified one leading exchange and the other three methods also chose notably little exchanges. Comparing with Table 2, we observe that TriSNAR and TLASSO, which both choose little parameters as non-zero, have the better BIC and TLASSO, which does not identify any influential exchanges, even outperforms in terms of BIC. This gives reason to believe that a change in the market structure took place in that time period. The BIC's of TriSNAR and TLASSO are quite close to each other and this period was subject to market turbulences with known cases of market manipulation which motivates to undergo a deeper investigation of this time period. Due to this market situation being of special interest, we ran a separate analysis only featuring the crisis period

Figure 5: Chord diagrams of the first lags of TriSNAR in Q2 2016 until Q1 2017 for the Bitcoin exchanges.

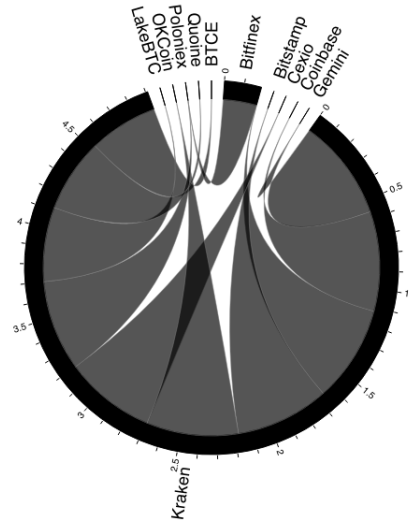
(a) TriSNAR lags 1 Q2 2016



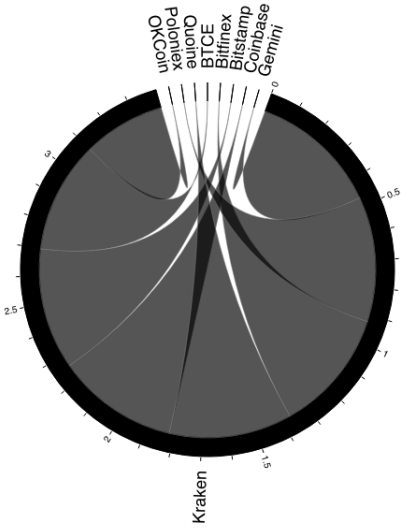
(b) TriSNAR lags 1 Q3 2016



(c) TriSNAR lags 1 Q4 2016



(d) TriSNAR lags 1 Q1 2017



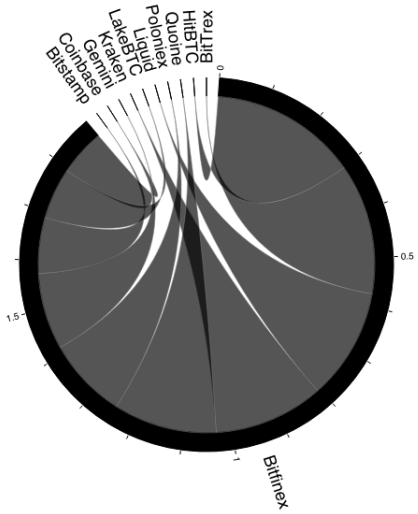
for the market evaluation. For that analysis, TriSNAR only identified one exchange as important whereas LASSO and TLASSO chose none, however SCAD choose two exchanges as relevant, compare Table 1. All methods only identify parameters in lag 1, hence we provide the chord diagrams for all methods first lags in Figure 6. TriSNAR only identified the exchange Bitfinex to be a leading exchange in this period whereas TLASSO, LASSO and SCAD are more dispersed however they also place high relevance on that particular exchange, since its parameters have a high magnitude. Comparing the BIC, see Table 2, TriSNAR achieved a much better BIC despite identifying much more parameters than the other three methods. This drives the question, why TriSNAR singled out Bitfinex as leading in this period. Griffin and Shams (2020) discovered fraudulent trading behaviour on the Bitfinex exchange via the BTC/USDT exchange pair. As TriSNAR suggests, this effect went further than fraudulent behaviour, the price series was manipulated so that it was leading the price discovery between all the exchanges.

With the start of 2018, we observe two quarters of high spread between the prices on the exchanges, however TriSNAR does not identify any influential exchanges for the price discovery. According to the BIC as reported in Table 2, no longer TriSNAR but TLASSO provides the best models for the entirety of 2018. TLASSO found no structure for Q1 2018, this means all parameters were zero. For Q2 2018, an unclear structure with parameters non-zero in all three lags was identified. This suggest that no clear structure was present in the network and no exchange acted as influencer in the price discovery network for these two quarters. In Q3 2018, the price spread between the exchanges becomes negligible and accordingly only autoregressive effects contribute to the price discovery, hence no cross-dependence between exchanges was found. In Q4 of 2018, TriSNAR identifies again only one exchange as relevant for the price discovery. TriSNAR identified only the exchange BitZ as influential, which is agreed upon by TLASSO, LASSO and SCAD as well, which place high magnitude on that exchanges parameters, see Figure 7. It is striking that the identified structure corresponds to the one in Q4 2017, when price manipulation took place on Bitfinex. To the best of our knowledge, there exists no research into the trading activities on this exchange in that particular time period. This puts at question if price manipulation might have taken place on BitZ as well. The CC comparison website CoinGecko publishes a trust score for exchanges and gives BitZ a 4 out of 10, which indicates that they consider the exchange as being of little trustworthiness. Following from this analysis, this might indicate a different case of market manipulation was found.

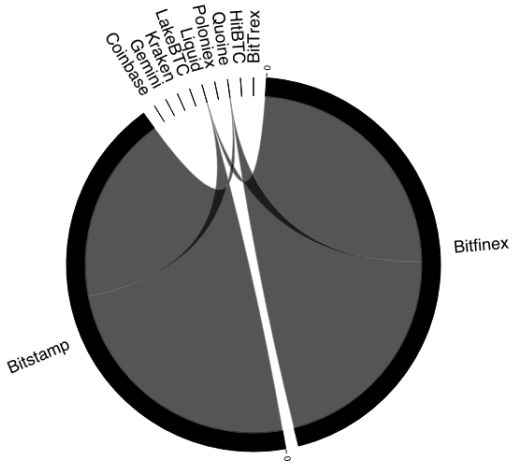
A related structure was also found in Q3 2015 for the exchange LakeBTC. Again TriSNAR identified only this exchange as influential and the other three methods place high magnitude on that exchange. However in that period LakeBTC was one the largest exchanges and influential in many periods, which might explain the influential role as well. Further the dominating influence of Kraken in 2016/2017 relates to a similar structure as

Figure 6: Chord diagrams of the first lags of TriSNAR, TLASSO, LASSO and SCAD in the 2017 subset for the Bitcoin exchanges. All methods only identified parameters in lag 1.

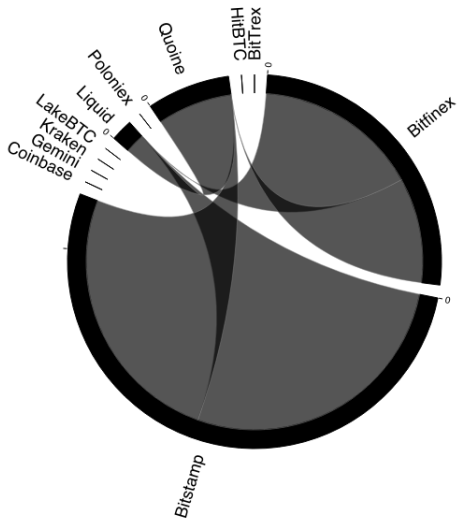
(a) TriSNAR lags 1



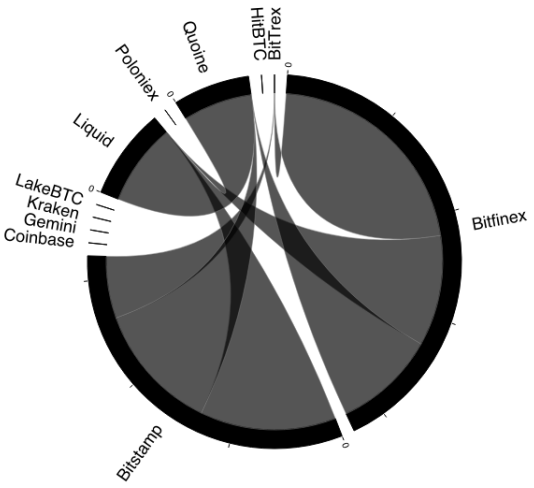
(b) TLASSO lags 1



(c) LASSO lags 1



(d) SCAD lags 1



discovered for Bitfinex during the 2017 market frenzy. However, the dominance of Kraken existed for an entire year which renders market manipulation unlikely and rather suggests a structural dominance.

4.5 2019 - 2021: Improving price discovery network efficiency

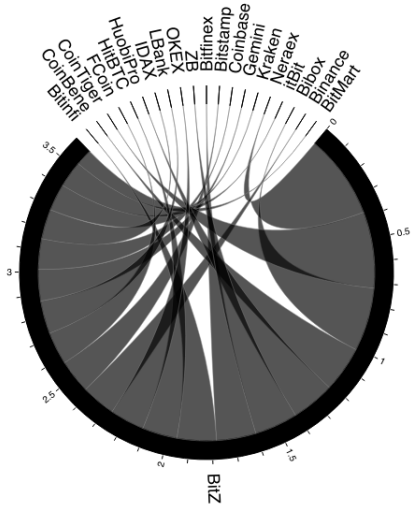
From 2019 onwards, we observe a low spread between the pricing series on the exchanges, compare Table 1, and TriSNAR usually finds none or only a few exchanges to be influential and the other methods support this observation. Taking into account Table 2, we observe that mostly TLASSO estimates the model with the best BIC. This suggests that the previously observed influencer structure vanished, since otherwise TriSNAR would receive the better BIC and identify leading exchanges. Provided that the spread between the pricing series is not too large and TLASSO usually provides the best model, this indicates that the pricing network has no leading exchanges in this time period which suggests a lower risk to the network because the market is more efficient compared to before. In Q3 2019 and Q1 2020, TLASSO gets outperformed by TriSNAR. In both quarters a low spread between the exchanges can be observed. However TriSNAR does not identify a clear structure either, which underscores the observation that the previously influential exchanges are no longer influential, which leads to an improved market efficiency.

4.6 2021 market frenzy: Continuing price discovery network efficiency

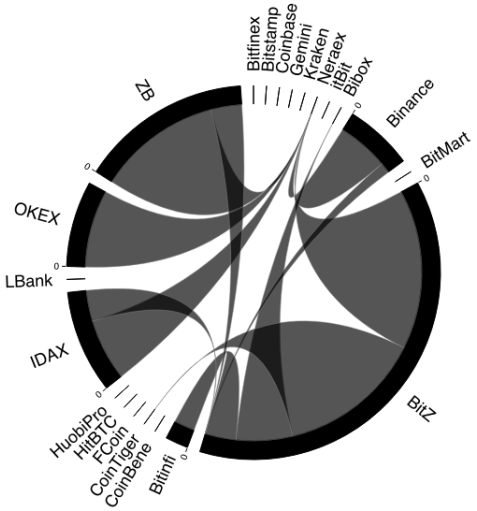
Even in Q1 of 2021 the connectivity between the exchanges remained similar as in 2019 and 2020, hence no influential exchanges were detected and again TLASSO performed best by BIC, compare Table 2. This is a surprising result considering that the spread between the exchanges is large in that quarter, Table 1, however it appears that the pricing network between the exchanges remained efficient. It is also remarkable that the pricing network stayed efficient considering Q1 of 2021 covers the time period of another market frenzy of Bitcoin in which the price reached various all-time-highs within a few weeks. Though it appears that the market situation was very different in this time period than it was during the market turbulences of 2017. Due to the special market situation in this time period, we also performed an analysis focusing on the time period of the market turbulences. The results indicate also in this subset analysis, that the market remained efficient, since no influential exchanges were detected and SCAD achieved the best BIC, compare Table 1 and 2. This result is in line with the ones from the quarter-by-quarter analysis.

Figure 7: Chord diagrams of the first lags of TriSNAR, TLASSO, LASSO and SCAD in the Q4 2018 for the Bitcoin exchanges. All methods only identified parameters in lag 1.

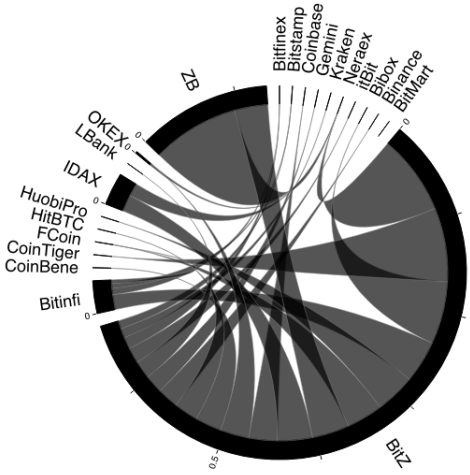
(a) TriSNAR lags 1



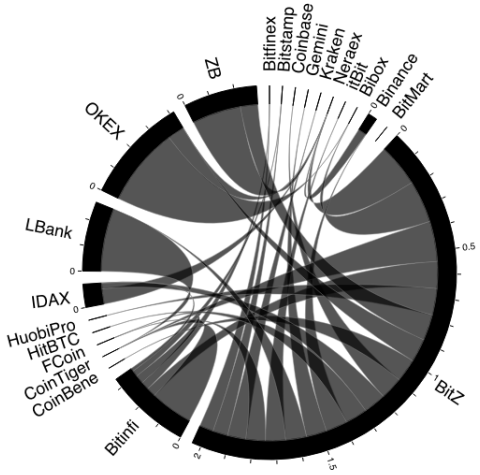
(b) TLASSO lags 1



(c) LASSO lags 1



(d) SCAD lags 1



5 Synthetic data experiments

We investigate the finite-sample performance of the proposed TriSNAR estimator with synthetic data. We consider various scenarios, ranging from simple cases with only one active lag to relatively complex cases with mixed lag, group and individual sparsity. We evaluate the ability to detect sparsity, the accuracy of the parameter estimation and prediction, as well as the runtime. We compare TriSNAR, derived with the active-set (TriSNAR$_G$) and approximating algorithm (TriSNAR$_A$), see Appendix I for the details of the algorithms, with a number of competing models. Two of the models, namely, LASSO and SCAD, only penalize for the individual parameters, while one other model, 'tapered LASSO' (TLASSO), Nicholson et al. (2014), penalizes for the lag structure by tapering off the effect of the individual parameter penalization. Consequently, it regularizes the lags and individual parameters. The estimation is implemented in the BigVAR package, Nicholson et al. (2019).

In the TriSNAR$_A$ optimization, we fix the identified lags, groups and parameters and activate FISTA after $m_1 = 5$ iterations, which provided frequently similar results in the synthetic data study. This value may not be sufficient for a different experiment or a different dataset. For dimension $d = 100$, we only derive the results for the approximating algorithm due to a slow optimization speed in high dimensions. We compare the computational time in the synthetic data study and also provide a detailed comparison in Appendix I.1. The code used in this paper as well as files containing the exact settings required for a replication of the synthetic data study are available at GitHub.

5.1 Set up

We consider networks with $d \in \{10, 20, 50, 100\}$ time series and lengths of $T \in \{100, 200, 500, 1,000\}$ data points. As we fix the active parameter matrices up to $p = 3$ lags, we need to derive 300, 1200, 7500 and 30000 unknown parameters of the adjacency matrices $\boldsymbol{A}_k$, $k = 1, \cdots, 3$. We design 6 model specifications and refer to each scenario by an abbreviation. We assign the capital letter D for specifications with a diagonal parameter matrix and the capital letter M for a medium persistent case containing group- and individual sparsity. A digit postfixed to the abbreviation indicates the number of active lag(s) in the specification. For example, a NAR model with the first and third lag being active with medium persistence is referred to as M1/M3. The details are listed below.

- *D1*: The first lag is a diagonal matrix with all diagonal parameters being 0.5. The other two lags are zero. In other words, there only exists autocorrelation in the network.
- *D2*: Similar specification as D1, except the active parameters appear in lag 2. The

first and third lags are non-active.

- *M1*: The first lag is active, with all diagonal parameters being of magnitude 0.5. There are 4 active columns $(1, 3, 7, 10)$ with alternating values of 0 and 0.15. The companion matrix of this specification has the largest eigenvalue of 0.75, yielding a medium persistence. The simultaneous evaluation of the lag, group and individual effects are required.

- *M2*: Similar to specification M1, except the active parameters occur in lag 2.

- *M1/M3*: Two lags are active, namely, the first and third lags. Both have the diagonal parameters of 0.2. The columns $1, 3, 7, 10$ are active with alternating values 0 and 0.1. The companion matrix of the specification has the largest eigenvalue of 0.87, indicating a medium persistence.

- *NS1*: We also consider a non-sparse specification, where only the first lag is active, yet there is no group and individual sparsity in the active matrix. The magnitude of the parameter decays exponentially away from the diagonal. It starts at 0.4 on the diagonal, and the off-diagonal parameters take on the values resulting from the formula $A_{i,j} = (-1)^{|i-j|}0.4^{|i-j|+1}$. In other words, all parameters are active; however, those far from the diagonal become quite small.

For specifications featuring only individual effects, e.g., NS1, the classic regularizations LASSO and SCAD are expected to perform well since both models are designed for such settings. When group effects are added, e.g., M1 and M2, TriSNAR is expected to excel since its three-layer design is appropriate, whereas the other models do not consider group effects. In the complex case of, e.g., M1/M3 with lag, group and individual effects present, TriSNAR and TLASSO should behave well as both regularize the lag and individual effects, though TriSNAR has the advantage of also considering the group effects.

In the data generation, the innovations are assumed to be i.i.d. Gaussian with $\epsilon \sim N(0, I_d)$. In all, there are $4(d) \times 4(T) \times 6(specifications) = 96$ experiments. For each experiment, we generated 100 synthetic datasets and computed the average performance. Moreover, we split each dataset into a training, validation and testing dataset, each with length T. We estimated the models for all combinations of $\lambda_1, \lambda_2,$ and λ_3-sequences on the training dataset. The λ_1, λ_2, λ_3-sequences are generated from deriving the values first for which all parameters are set to 0. From this value a halving sequence is created for each λ until 0 is reached. The best performing model was evaluated based on the validation dataset with BIC. The resulting model was then implemented for the test sample for its forecasting accuracy.

5.2 Evaluation Criteria

The performance was evaluated in three aspects: pattern, accuracy and speed.

- To evaluate the pattern identification, we computed the False Negative (FN) and False Positive (FP) rates on the estimated sets. FN refers to active set's being falsely identified as null, namely, under-detection or overly sparse. FP refers to the set's being wrongly identified as active, namely, overdetection or overly dense. It is natural that the lower these two measures are, the better the performance. Given the three-layer sparsity, there are then 6 metrics: FN.l and FP.l for lags, FN.g and FP.g for groups, and FN.e and FP.e for individual elements. In the case of perfect detection, namely, all 6 metrics are zero, we conclude that the true pattern was identified.

- Accuracy is measured using the Mean Absolute Error (MAE). Again, there are three different metrics. MAE.para refers to the estimation accuracy, computed based on the difference between the true and estimated parameters. MAE.res refers to the prediction accuracy, which is calculated based on the residuals between the true values of the time series and the predicted values based on the model. In other words, it evaluates in-sample on the training dataset. MAFE.res refers to the forecast error, an out-of-sample measure based on the testing dataset. In all the accuracy metrics, a low value indicates good accuracy.

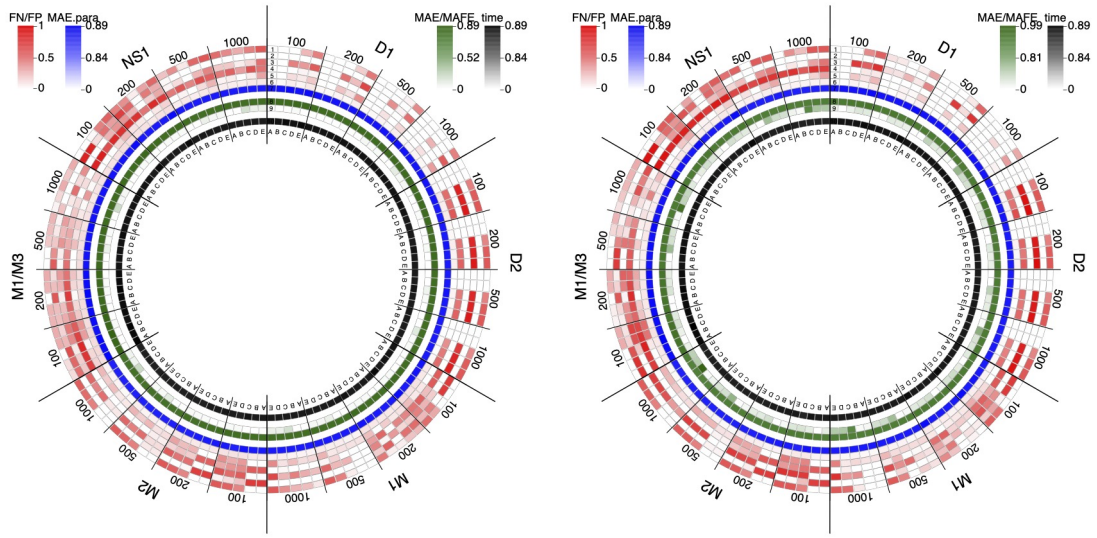
- Time/combination is an indicator to measure the speed. We report the time in seconds it took on average to derive the model per combination of penalization sequences.

5.3 Roseplots: Summary of the Results

The Figures 8 and 9 summarize the performance of TriSNAR and competing estimators along with the 10 measurements in the 96 experiments, separated according to model specification. We provide the detailed results in Tables in Appendix IV. Each of the 4 roseplots shows the performance of the synthetic data study for a given dimension for all 6 scenarios, all number of observations and the 10 evaluation criteria. Each roseplot is separated into 6 sections, one for each scenario. Within each section, 4 subsections are assigned for the number of observations: $T = \{100, 200, 500, 1000\}$. For each of these subsections, 5 columns of rectangles are provided, named A, B, C, D, E. The naming convention refers to TriSNAR$_G$ (A), TriSNAR$_A$ (B), TLASSO (C), SCAD (D) and LASSO (E).

The entire circle has 10 tracks, each of which represents another evaluation criterion. The most outer track is referred to as 1, and the most inner track is referred to as 10. The 6 most outer tracks are for the False Negative and False Positive criteria: FN.l (1), FP.l (2), FN.g (3), FP.g (4), FN.e (5), FP.e (6). The FN and FP rates vary between 0 and 1,

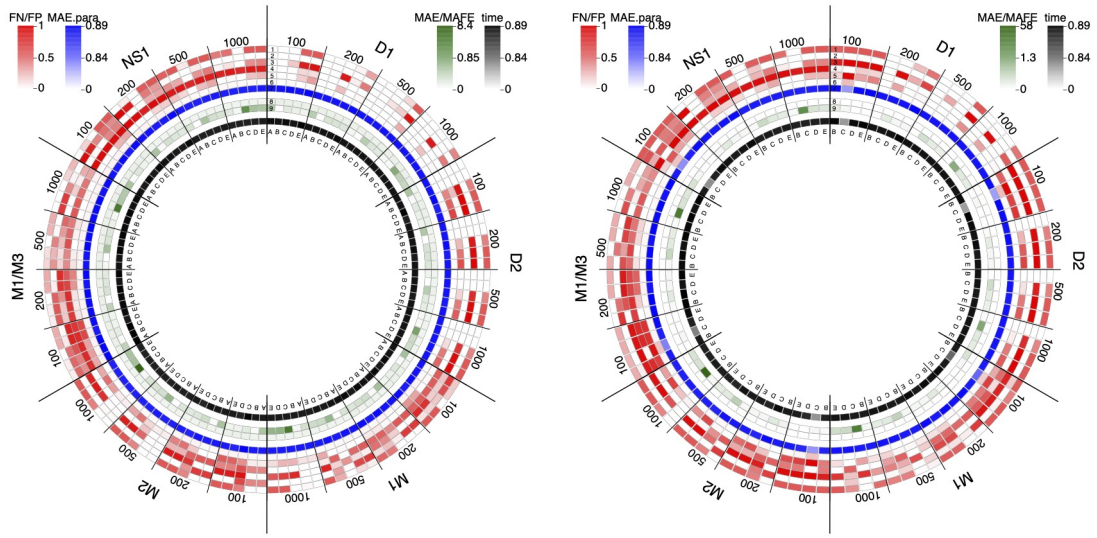
Figure 8: Roseplot for $d = 10$ (left) and $d = 20$ (right)



whereas the color palette goes from white (0) to red (1). No False Negatives and no False Positives are the best possible outcomes; hence, the more white or shallow red the rectangles are, the better. Track 7 reports the MAE.para with a color palette from white (0) to blue (maximum value of MAE.para for the dimension d under evaluation). Again, the lower the value, the better; hence, white or shallow blue rectangles are preferable. The MAE.res is reported via track 8 and the MAFE.res via track 9. The color palette goes from white (0) to green (maximum value of MAE.res/MAFE.res for the dimension d under evaluation). Since these evaluation criteria reflect the error terms, the smaller the values are, the better. Thus, white and shallow green is preferable. The most inner track, 10, reports the runtime per combination of λ values. The color palette ranges from white (0) to black (maximum value of runtime per comparison for the dimension d under evaluation). Certainly, a faster runtime of the code is preferred; hence, white and gray rectangles are better.

It can be observed that the rectangles associated with TriSNAR$_G$ and TriSNAR$_A$ with $d = \{10, 20, 50\}$ for the FN and FP values are always white for D1 and D2. For $d = 100$, the rectangles become all white for $T \geq 200$. However, all the other models show mostly dark red rectangles on tracks 2, 4, and 6, which reflect the FP rates. Hence, the competing estimators overparametrize the models; therefore, they do not provide the true model. For D1 and a higher number of observations, SCAD also returned the true model, although only TriSNAR identified the true structure in both cases and for a small number of observations. For scenarios M1 and M2, overall, TriSNAR$_G$ and TriSNAR$_A$ performed best in terms of uncovering the true structure. For $T = 100$, both models provided a too sparse model, while the competing estimators selected mostly the incorrect parameters and the incorrect structure. We observe that all rectangles of C, D, and E are shaded red, which indicates that incorrect parameters were chosen, whereas the true parameters were not included in

Figure 9: Roseplot for $d = 50$ (left) and $d = 100$ (right)



the model. Hence, the TriSNAR solution is preferable even though the solution was too sparse. For $T \geq 200$, TriSNAR$_G$ and TriSNAR$_A$ became continuously better, whereas estimators C, D, and E overparameterized. For M1/M3 and NS1 scenarios, exploring the underlying structure became more difficult for all involved estimators. It can be observed that the TriSNAR estimators gave a more accurate estimate, which was inferred from the observation that the respective rectangles are white or have more shallow red compared to the competing estimators. TriSNAR$_G$ and TriSNAR$_A$ also overparameterized, however, less intense than the competing estimators.

Considering the MAE.para, track 7, it can be observed that both TriSNAR estimators provided more accurate parameter estimations in most cases. When they did not outperform, the performance was similar to the other models. This result corresponds with the observation that TriSNAR$_G$ and TriSNAR$_A$ provided more accurate solutions in terms of identifying the underlying model structure.

Tracks 8 and 9 provided the results for MAE.res and MAFE.res. For all models within one subsection, the results for MAE.res were comparable however for MAFE.res the TriSNAR$_G$ and TriSNAR$_A$ models usually outperform. Hence their prediction accuracy is better. This infers that all models provide similar model performance in-sample, though the out-of-sample (prediction) performance of TriSNAR$_G$ and TriSNAR$_A$ is much better than for the competing methods. Further, as was discussed before, models C, D, and E tend to overparameterize the models. In contrast, TriSNAR$_G$ and TriSNAR$_A$ can identify the true model structure more often while providing better prediction accuracy.

Track 10 compares the runtime per combination of λ sequences. We observe that the TriSNAR estimators had the fastest runtime, whereas the TLASSO model (C) frequently

had a much longer runtime than the other estimators. SCAD and LASSO performed comparably in terms of runtime, and they were derived faster than TLASSO.

The good performance of TriSNAR in the synthetic data study for various amounts of finite data calls for an investigation of the asymptotic properties of TriSNAR. Under the assumption of a fixed dimensionality, we study the asymptotic properties of TriSNAR in Appendix II. We show that TriSNAR has the sparsity and oracle property under the outlined assumptions.

The good performance of TriSNAR over all evaluation metrics was possible due to its properties. Due to its structure TriSNAR is able to detect influencer structures faster than competing methods, whereas it remains flexible when such structures do not exist. Also its ability to regularize the temporal structure contributes to its better performance. In summary, TriSNAR outperformed in the majority of the scenarios and cases, illustrating its applicability to large-scale networks.

6 Conclusion

We study the price discovery network between Bitcoin exchanges with a focus upon identification of influential exchanges, market efficiency and market manipulation identification. We identified various influential exchanges before 2018, strikingly the exchange Kraken remained influential for an entire year, namely from April 2016 until March 2017. We relate this observation to its popularity back then and that Kraken is headquartered in the USA whereas many other exchanges were back then based in less regulated jurisdictions. We found that the efficiency of the BTC price discovery network decreased from 2015 until the 2017 market frenzy, however it improved from 2018 onwards. This observations stems from the existence of influential exchanges during the period 2015-2017 and evidence of structural inefficiency relating to the detection of Kraken as an influential exchange for an entire year. However the situation improved from 2018 onwards, which relates to the detection of less, up to none, influential exchanges. Even during the market frenzy of 2021, no influential exchanges were detected hence the improved market efficiency prevailed. This also provides evidence that the market frenzies of 2017 and 2021 were structurally different from each other. In terms of fraud detection, we identified a single influential exchange during the market frenzy of 2017, namely Bitfinex, which was found to be the platform for market manipulation during that time. We identified similar patterns a few more times, notably for the exchange BitZ one year later, which calls at question if it might have been used for fraudulent behaviour as well. Other cases involved Kraken and LakeBTC in the time period 2015-2017, however due to their popularity in this time period, the observation is more likely related to actual influence on the price discovery network rather market ma-

nipulation. For the study we propose a model, called TriSNAR, to identify the influential exchanges in the network. We study the asymptotic properties of TriSNAR and show in extensive synthetic data studies that it outperforms competing models in a finite sample setting. Compared to the other methods, TriSNAR excels in terms of accuracy, runtime and its ability to discover the influencers in a network structure. This study improves the understanding of the price discovery in the Bitcoin market and proposes a model, TriSNAR, to study such multi-market networks which outperformed competing methods.

References

Alexander, C., Choi, J., Park, H., and Sohn, S. (2020). "BitMEX bitcoin derivatives: Price discovery, informational efficiency, and hedging effectiveness". *Journal of Futures Markets* 40.1, pp. 23–43.

Antoniadis, A. (1997). "Wavelets in Statistics: A Review". *Journal of the Italian Statistical Society* 6.2, pp. 97–130.

Asgharian, H., Hess, W., and Liu, L. (2013). "A spatial analysis of international stock market linkages". *Journal of Banking & Finance* 37.12, pp. 4738–4754.

Bańbura, M., Giannone, D., and Reichlin, L. (2010). "Large Bayesian Vector Autoregressions". *Journal of Applied Econometrics* 25.1, pp. 71–92.

Barigozzi, M. and Brownlees, C. (2019). "NETS: Network estimation for time series". *Journal of Applied Econometrics* 34.3, pp. 347–364.

Basu, S., Li, X., and Michailidis, G. (2019). "Low Rank and Structured Modeling of High-Dimensional Vector Autoregressions". *IEEE Transactions on Signal Processing* 67.5, pp. 1207–1222.

Basu, S. and Michailidis, G. (2015). "Regularized Estimation in Sparse High-Dimensional Time Series Models". *The Annals of Statistics* 43.4, pp. 1535–1567.

Beck, A. and Teboulle, M. (2009). "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". *SIAM Journal on Imaging Sciences* 2.1, pp. 183–202.

Bonaldi, P., Hortaçsu, A., and Kastl, J. (2015). *An Empirical Analysis of Funding Costs Spillovers in the EURO-zone with Application to Systemic Risk*. Working Paper 21462. National Bureau of Economic Research.

Capponi, A. and Jia, R. (2021). "The Adoption of Blockchain-based Decentralized Exchanges". arXiv: `2103.08842 [q-fin.TR]`.

Chen, C. Y.-H., Härdle, W. K., and Klochkov, Y. (2021). "SONIC: SOcial Network analysis with Influencers and Communities". *Journal of Econometrics*.

Chen, G.-m., Firth, M., and Meng Rui, O. (2002). "Stock market linkages: Evidence from Latin America". *Journal of Banking & Finance* 26.6, pp. 1113–1141.

Chen, H., Choi, P. M. S., and Hong, Y. (2013). "How smooth is price discovery? Evidence from cross-listed stock trading". *Journal of International Money and Finance* 32, pp. 668–699.

Chen, Y., Trimborn, S., and Zhang, J. (2018). "Discover Regional and Size Effects in Global Bitcoin Blockchain via Sparse-Group Network AutoRegressive Modeling". available at SSRN ID 3245031.

Cong, L. W., Li, X., Tang, K., and Yang, Y. (2021). "Crypto wash trading". *arXiv preprint arXiv:2108.10984*.

Creal, D., Koopman, S. J., and Lucas, A. (2013). "Generalized Autoregressive Score Models with Applications". *Journal of Applied Econometrics* 28.5, pp. 777–795.

Davis, R. A., Zang, P., and Zheng, T. (2016). "Sparse Vector Autoregressive Modeling". *Journal of Computational and Graphical Statistics* 25.4, pp. 1077–1096.

de Paula, Á. (2017). "Econometrics of Network Models". *Advances in Economics and Econometrics: Eleventh World Congress*. Ed. by B. Honoré, A. Pakes, M. Piazzesi, and L. Samuelson. Cambridge University Press, pp. 268–323.

Dees, S., di Mauro, F., Pesaran, M. H., and Smith, L. V. (2007). "Exploring the International Linkages of the EURO Area: A Global VAR Analysis". *Journal of Applied Econometrics* 22.1, pp. 1–38.

Diebold, F. X. and Yılmaz, K. (2014). "On the Network Topology of Variance Decompositions: Measuring the Connectedness of Financial Firms". *Journal of Econometrics* 182.1. Special Issue on Causality, Prediction, and Specification Analysis: Recent Advances and Future Directions, pp. 119–134.

Eun, C. S. and Shim, S. (1989). "International Transmission of Stock Market Movements". *Journal of Financial and Quantitative Analysis* 24.2, pp. 241–256.

Fan, J. and Li, R. (2001). "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties". *Journal of the American Statistical Association* 96.456, pp. 1348–1360.

Gagnon, L. and Karolyi, G. (2013). "International Cross-Listings". *The Evidence and Impact of Financial Globalization*. Ed. by G. Caprio, T. Beck, S. Claessens, and S. L. Schmukler. San Diego, CA, USA: Academic Press, pp. 155–180.

Gagnon, L. and Karolyi, G. A. (2010). "Multi-market trading and arbitrage". *Journal of Financial Economics* 97.1, pp. 53–80.

Ghosh, S., Khare, K., and Michailidis, G. (2019). "High-Dimensional Posterior Consistency in Bayesian Vector Autoregressive Models". *Journal of the American Statistical Association* 114.526, pp. 735–748.

Griffin, J. M. and Shams, A. (2020). "Is Bitcoin Really Untethered?" *The Journal of Finance* 75.4, pp. 1913–1964.

Guo, L., Tao, Y., and Härdle, W. K. (2019). *A Dynamic Network Perspective on Cryptocurrencies*. SSRN Paper ID 3185594. Rochester, NY, USA: Social Science Research Network.

Halling, M., Moulton, P. C., and Panayides, M. (2013). "Volume Dynamics and Multimarket Trading". *Journal of Financial and Quantitative Analysis* 48.2, pp. 489–518.

Härdle, W. K., Wang, W., and Yu, L. (2016). "TENET: Tail-Event driven NETwork Risk". *Journal of Econometrics* 192.2. Special Issue on Innovations in Multiple Time Series Analysis, pp. 499–513.

Kline, B. and Tamer, E. (2017). "Econometric Analysis of Models with Social Interactions". available at semanticscholar.org.

Kock, A. B. and Callot, L. (2015). "Oracle Inequalities for High Dimensional Vector Autoregressions". *Journal of Econometrics* 186.2. Special Issue on High Dimensional Problems in Econometrics, pp. 325–344.

Lin, J. and Michailidis, G. (2017). "Regularized Estimation and Testing for High-Dimensional Multi-Block Vector-Autoregressive Models". *Journal of Machine Learning Research* 18.1, pp. 4188–4236.

Lin, W.-L., Engle, R., and Ito, T. (1994). "Do bulls and bears move across borders? International transmission of stock returns and volatility". *The Review of Financial Studies* 7.3, pp. 507–538.

Makarov, I. and Schoar, A. (2019). "Price Discovery in Cryptocurrency Markets". *AEA Papers and Proceedings* 109, pp. 97–99.

Makarov, I. and Schoar, A. (2020). "Trading and arbitrage in cryptocurrency markets". *Journal of Financial Economics* 135.2, pp. 293–319.

Nelson, C. R. and Plosser, C. R. (1982). "Trends and random walks in macroeconmic time series: Some evidence and implications". *Journal of Monetary Economics* 10.2, pp. 139–162.

Nicholson, W., Matteson, D., and Bien, J. (2019). *BigVAR: Dimension Reduction Methods for Multivariate Time Series*. R package version 1.0.4.

Nicholson, W., Matteson, D., and Bien, J. (2017). "VARX-L: Structured Regularization for Large Vector Autoregressions with Exogenous Variables". *International Journal of Forecasting* 33.3, pp. 627–651.

Nicholson, W. B., Wilms, I., Bien, J., and Matteson, D. S. (2014). "High Dimensional Forecasting via Interpretable Vector Autoregression". arXiv:1412.5250 [stat].

Pesaran, M. H., Schuermann, T., and Weiner, S. M. (2004). "Modeling Regional Interdependencies Using a Global Error-Correcting Macroeconometric Model". *Journal of Business & Economic Statistics* 22.2, pp. 129–162.

Rapach, D. E., Strauss, J. K., and Zhou, G. (2013). "International Stock Return Predictability: What Is the Role of the United States?" *The Journal of Finance* 68.4, pp. 1633–1662.

Said, S. E. and Dickey, D. A. (1984). "Testing for unit roots in autoregressive-moving average models of unknown order". *Biometrika* 71.3, pp. 599–607.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). "A Sparse-Group Lasso". *Journal of Computational and Graphical Statistics* 22.2, pp. 231–245.

Skripnikov, A. and Michailidis, G. (2019). "Regularized joint estimation of related vector autoregressive models". *Computational Statistics & Data Analysis* 139, pp. 164–177.

Song, S. and Bickel, P. J. (2011). "Large Vector Autoregressions". arXiv:1106.3915 [stat.ML].

Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288.

Tsay, R. S. (2016). "Some Methods for Analyzing Big Dependent Data". *Journal of Business & Economic Statistics* 34.4, pp. 673–688.

Wang, H., Li, G., and Tsai, C.-L. (2007). "Regression Coefficient and Autoregressive Order Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.1, pp. 63–78.

Zhu, X., Pan, R., Li, G., Liu, Y., and Wang, H. (2017). "Network Vector Autoregression". *The Annals of Statistics* 45.3, pp. 1096–1123.

Zou, H. (2006). "The Adaptive Lasso and Its Oracle Properties". *Journal of the American Statistical Association* 101.476, pp. 1418–1429.

I Algorithm - Appendix

We develop two algorithms to implement the estimation: an active-set algorithm based on coordinate-wise descent and an approximating algorithm with early termination for fast solutions. The latter switches from the active-set approach after a given number of iterations and continues searching for a solution with the FISTA algorithm, Beck and Teboulle (2009), solely for the active parameters already discovered.

In the following, we describe the algorithms in more detail. The vector Y_t without the jth process is denoted by $Y_{t,-j}$, and recall that $Y_{t,j}$ represents the jth process. Define $sort(\cdot)$ as the operator that sorts the variables in decreasing order. We use an active-set algorithm to sequentially evaluate the three-layer parameters with the order of lags, groups and individuals. The algorithm is initialized with completely sparse parameter matrices, meaning that all parameters are set equal to 0. Since we assume that the parameter matrix is sparse, this can be considered an appropriate starting point. First, we sort the lags, groups and individual parameters according to the proportion of variance unexplained by them. The sorting ensures that, the algorithm optimizes first the parameters that explain more of the variability of the system. In the algorithm, we iterate over active lags in an outer optimization loop. We iterate over the active groups in each of the identified active lags in a similar manner. The active individual elements are identified and optimized only from the active group sets. In each iteration, we construct the residuals ϵ_k, $\epsilon_{k;-j}$ and $\epsilon_{k;ij}$, reflecting the unexplained variance, based on which we optimize the parameters for the active sets. While iterating, non-active lags, groups and individual parameters are removed from further analysis.

The implementations are formulated as Algorithm **TriSNAR.lag**, Algorithm **TriSNAR.group** and Algorithm **TriSNAR.individual**.

1. TriSNAR.lag is the first outer loop algorithm. It evaluates the tuning parameter λ_1 to identify the lag parameter matrix carrying sufficient information. We sort the matrices in decreasing order according to the explained variance as reflected in the residuals ϵ_k. In each iteration step (m_1), $\boldsymbol{A}_k$ with little or no explanatory power is forced to be 0. Otherwise, with a sufficiently large explained contribution to the variance, i.e., $\epsilon_k > d^2\lambda_1$, we continue to estimate the lag parameter matrix with Algorithm TriSNAR.group.

2. TriSNAR.group evaluates the groups on the sequence λ_2. Similarly, we order them according to the explained variance, and the algorithm is iterated with the residual $\epsilon_{k;-j}$. In each iteration step (m_2), $A_{k;j}$ is set to 0 in the case of little or no explanatory power. Otherwise, if $\epsilon_{k;-j} > d\lambda_2$, we continue the implementation with Algorithm TriSNAR.individual.

3. TriSNAR.individual is used to optimize the individual parameters inside an active group. It is a coordinate-wise descent optimization under the sequence λ_3 and with residual $\epsilon_{k;ij}$ according to estimator (5). In each iteration step (m_3), the contribution to the variance is evaluated. In case there is little or no explanatory power, $A_{k;ij}$ is set to 0. Otherwise, the non-zero parameter is estimated following the case differentiation of the estimator (5).

4. The algorithms are repeated with iteration steps m_1, m_2, m_3 until all parameter matrices have converged.

The implementation depends on the hyperparameters η_1, η_2, η_3, which are user specified. The parameter b for the estimator of the individual parameters can also be set as a sequence. However, this part of the estimator corresponds to SCAD; hence, we follow the recommendation of Fan and Li (2001) and set $b = 3.7$. The regularization sequences remain to be selected, i.e., the values of the tuning parameters λ_1, λ_2, and λ_3. Usually, cross-validation is used to determine the sequence. However, due to the time dependence in our model, cross-validation is not very suitable. We choose the tuning parameters using the out-of-sample BIC. The use of information criteria for the evaluation is consistent with Bańbura et al. (2010), Song and Bickel (2011), Nicholson et al. (2017) and Chen et al. (2018). The run time depends on the size of the sequence of tuning parameters λ_1, λ_2 and λ_3. Naturally, a more granular penalization sequence leads to a longer runtime of the optimization procedure. In our case, it is a halving sequence approaching 0 for the individual parameter penalization (λ_3) and a diminishing sequence by 1/5 for the group (λ_2) and lag regularization (λ_1), also approaching 0. This default setting provides stable performance in the synthetic data study.

For very high dimensions or difficult specifications, we also propose an approximating algorithm that can find a faster solution but may lead to a local optimizer. In the approximating algorithm, the described procedure interrupts after a specified number of iterations (s). At this point, the by then identified lagged matrices and groups are considered active, and the remaining ones are set to 0. Then, we apply the FISTA algorithm to the parameters that are considered active. FISTA is not applied in the active-set algorithm because the convergence takes longer over the active sets. Coordinate-wise descent, delivering similar parameter estimates, is beneficial because an easier parameter-updating step improves the speed due to iterating over the sets of parameters. However, FISTA performs well for the approximating algorithm because it can optimize faster over all sets at once instead of iterating over them, which improves the estimation time of the parameters. In the numerical analysis, the results derived with the active-set optimization algorithm are denoted by TriSNAR$_G$ and the approximate result by TriSNAR$_A$.

Algorithm 1 : TriSNAR.lag

Input: Data Y_t for all $t = 1, \ldots, T$
Output: Adjacency matrix $\boldsymbol{A}$

Initialization $\boldsymbol{A} = 0$, $m_1 = 1$
for $k = 1, \ldots, p$ **do**
  $\epsilon.lag_k = \sqrt{\sum_{t=p+1}^{T}(Y_{t-k}^{\top}(Y_t - \sum_{l=1\setminus k}^{p} \boldsymbol{A}_l Y_{t-l}))^2}$
end for
$order.lag = sort(\{\epsilon.lag_k\}_{k=1}^{p})$

$\boldsymbol{A}^{(m_1)} = \boldsymbol{A}$; $\boldsymbol{A}^{(m_1-1)} = \boldsymbol{A} + 1$
while $\text{vec}\{\boldsymbol{A}^{(m_1)} - \boldsymbol{A}^{(m_1-1)}\} < \eta_1$ and $m_1 \leq s$ **do**
  for $k \in order.lag$ **do**
    $m_2 = 1$
    $\epsilon_k = \sqrt{\sum_{t=p+1}^{T}(Y_{t-k}^{\top}(Y_t - \sum_{l=1\setminus k}^{p} \boldsymbol{A}_l^{(m_2)} Y_{t-l}))^2}$
    if $\epsilon_k \leq d^2\lambda_1$ **then** $\boldsymbol{A}_k^{(m_2)} = 0$
    else
      $\boldsymbol{A}_k^{(m_2)} = \boldsymbol{A}_k$; $\boldsymbol{A}^{(m_2-1)} = \boldsymbol{A}_k + 1$
      while $\boldsymbol{A}_k^{(m_2)} - \boldsymbol{A}_k^{(m_2-1)} < \eta_2$ **do**
        TriSNAR.group$(\{Y_t\}_{t=1}^{T}, \boldsymbol{A}_k^{(m_2)})$
        $m_2 = m_2 + 1$
      end while
    end if
    $m_1 = m_1 + 1$
  end for
end while

Algorithm 2 : TriSNAR.group

Input: Data Y_t for all $t = 1, \ldots, T$; $\boldsymbol{A}_k$
Output: Adjacency matrix $\boldsymbol{A}_k$

for $j = 1, \ldots, d$ **do**
  $\epsilon.group_j = \sqrt{\sum_{t=p+1}^{T} (Y_{t-k,-j}^{\top} (Y_t - \sum_{l=1\setminus k}^{p} \boldsymbol{A}_l Y_{t-l})_{-j})^2}$
end for
$order.group = sort(\{\epsilon.group_j\}_{j=1}^{d})$

for $j \in order.group$ **do**
  $m_3 = 1$
  $\epsilon_{k;-j} = \sqrt{\sum_{t=p+1}^{T} (Y_{t-k,-j}^{\top} (Y_t - \sum_{l=1\setminus k}^{p} \boldsymbol{A}_l^{(m_3)} Y_{t-l})_{-j})^2}$
  if $\epsilon_{k;-j} \leq d\lambda_2$ **then** $A_{k;j}^{(m_3)} = 0$
  else
    $A_{k;j}^{(m_3)} = A_{k;j}$; $A_{k;j}^{(m_3-1)} = A_{k;j} + 1$
    while $A_{k;j}^{(m_3)} - A_{k;j}^{(m_3-1)} < \eta_3$ **do**
      TriSNAR.individual$(\{Y_t\}_{t=1}^{T}, A_{k;j}^{(m_3)})$
      $m_3 = m_3 + 1$
    end while
  end if
end for

Algorithm 3 : TriSNAR.individual

Input: Data Y_t for all $t = 1, \ldots, T$; $A_{k;j}$
Output: Adjacency matrix $A_{k;j}$

for $i = 1, \ldots, d$ **do**
  $\epsilon_{k;ij} = \sqrt{\sum_{t=p+1}^{T} (Y_{t-k,j}^{\top} (Y_{t,i} - \sum_{l=1\setminus k}^{p} A_{l,ij}^{(m_3)} Y_{t-l,j}))^2}$
  if $|\epsilon_{k;ij}| \leq 2\lambda_3$ **then** $z = sgn(\epsilon_{k;ij})(|\epsilon_{k;ij}| - \lambda_3)_+$
  else if $2\lambda_3 < |\epsilon_{k;ij}| \leq b\lambda_3$ **then** $z = \frac{(b-1)\epsilon_{k;ij} - sgn(\epsilon_{k;ij})b\lambda_3}{(b-2)}$
  else if $b\lambda_3 < |\epsilon_{k;ij}|$ **then** $z = \epsilon_{k;ij}$
  end if
  $A_{k;ij} = z / \sum_{t=p+1}^{T} (Y_{t-k,j}^{\top} Y_{t-k,j})$
end for

I.1 Comparison of derivation time

The results shown in Table 3 are deducted from the synthetic data study in section 5. The table shows the time the estimation took for each sample size $T \in \{100, 200, 500, 1000\}$ for TriSNAR$_G$. The results are compared with the synthetic datasets with $d = 10$ time series for 3, 5 and 7 lags in the algorithmic specifications. The computational time increases when more lags are involved; however, with a higher number of observations, the computation time mostly decreases. This is particularly true for 5 and 7 lags. One observes that the computational time for those models only involving the diagonal, *D1* and *D2*, is comparable to models involving network effects of medium persistence, *M1*, *M2*, *M1/M3*. Interestingly, the estimation time of the models is lower when the third lag is also active, meaning *M1. M3*, than for its peers with only the first or second lag active (*M1*, *M2*). Compared with the computational time with the competing models in the synthetic data study, it can be observed that TriSNAR$_{G/A}$ has a considerably lower computational time for each combination of λ's than the competing models; see, e.g., Table 10. The granularity of the λ sequences defines the performance. The lesser combinations, namely, a less granular grid of λ, reduce the overall runtime of the models and are therefore crucial for applying any of the reported methods.

Table 3: Average duration of derivation in seconds for synthetic data experiments with 100, 200, 500, 1000 observations and for 3, 5, 7 lags for the specifications D1, D2, M1, M1/M3, M2, NS1. For the sake of brevity only the experiments for 10 time series are shown, however the results for larger systems give comparable results.

	100			200			500			1000		
	3	5	7	3	5	7	3	5	7	3	5	7
D1	68	317	623	52	145	335	73	142	267	122	391	387
D2	67	255	722	50	121	252	69	159	314	107	211	364
M1	98	275	737	87	209	420	107	219	441	160	307	520
M1/M3	69	240	682	44	113	253	89	165	283	147	282	482
M2	88	275	749	72	172	351	102	224	397	143	300	527
NS1	99	284	777	93	239	506	121	240	410	176	347	627

II Theoretical Properties - Appendix

We derive the asymptotic properties of the estimator (5). Recall that the matrices $\boldsymbol{A}_k$ are assumed to have a sparse 'network' structure. The residual term $\epsilon_t = (\epsilon_{1,t}, \cdots, \epsilon_{d,t})^\top$ is a vector that is assumed to be independently and identically distributed with $\epsilon_t \sim (0, \Sigma)$. We assume that the model is stationary and ergodic, with all roots of the polynomial $\boldsymbol{I}_d - \sum_{k=1}^{p} \boldsymbol{A}_k Z^k$ lying outside the unit ball. Note that d is fixed during the investigation of the asymptotic properties. We assume the following regularity conditions hold:

1. The observations Y_t for all t are i.i.d. with probability density $f(Y, \boldsymbol{A})$. It shall hold

$$\mathrm{E}\left[\frac{\partial log f(Y, \boldsymbol{A})}{\partial A_{k;ij}}\right] = 0 \text{ for all } i, j = 1, \ldots, d \text{ and } k = 1, \ldots, p$$

and

$$I_{k_1 i_1 j_1, k_2 i_2 j_2} = \mathrm{E} - \frac{\partial^2 log f(Y, \boldsymbol{A})}{\partial A_{k_1;i_1 j_1} \partial A_{k_2;i_2 j_2}}$$

2. The Fisher Information matrix $I(\boldsymbol{A})$ is finite and positive definite at $\boldsymbol{A} = \mathcal{A}$ with $\mathcal{A}$ the true parameter matrix.

3. There exists an open subset ω in the parameter space Ω of $\boldsymbol{A}$ that contains the true parameter matrix $\mathcal{A}$. For almost all Y_t the density $f(Y, \boldsymbol{A})$ admits all third derivatives $\frac{\partial^3 log f(Y, \boldsymbol{A})}{\partial A_{i_1 j_1} \partial A_{i_2 j_2} \partial A_{i_3 j_3}}$ for all $\boldsymbol{A}$ in the open subset. There exist functions $M_{i_1 j_1, i_2 j_2, i_3 j_3}$ such that

$$\left|\frac{\partial^3 log f(Y, \boldsymbol{A})}{\partial A_{i_1 j_1} \partial A_{i_2 j_2} \partial A_{i_3 j_3}}\right| \leq M_{i_1 j_1, i_2 j_2, i_3 j_3}(Y) \text{ for all } \boldsymbol{A} \in \omega$$

whereas $m_{i_1 j_1, i_2 j_2, i_3 j_3} = \mathrm{E}\left[M_{i_1 j_1, i_2 j_2, i_3 j_3}(Y)\right] < \infty$.

Note that T will go to infinity which impacts the values of $\lambda_1, \lambda_2, \lambda_3$, hence we denote them $\lambda_{1,T}, \lambda_{2,T}, \lambda_{3,T}$ from here onwards. Denote $g_{max,T} = \max(\frac{\partial p_{\lambda_{1,T}, \lambda_{2,T}, \lambda_{3,T}}}{\partial A_{k;ij}} : A_{k;ij} \neq 0)$, which is the maximal regularization applied to any $A_{k;ij}$. $g_{max,T}$ will only take on the value 0 if $\lambda_{1,T}, \lambda_{2,T}, \lambda_{3,T} \to 0$. Also in case of a dense system, $g_{max,T}$ would be 0 but this contradicts the assumption of this study of sparse parameter matrices. The proofs to the results are given in detail in Appendix III.

Theorem 1 *Assume that the assumptions for model (2) hold. If* $\max\{\frac{\partial^2 p_{\lambda_{1,T}, \lambda_{2,T}, \lambda_{3,T}}}{\partial A^2_{k;ij}(A_{k;ij})} : A_{k;ij} \neq 0\} \to 0$*, then there exists a local maximizer* $\widehat{\boldsymbol{A}}$ *for (3) such that* $||\widehat{\boldsymbol{A}} - \boldsymbol{A}_0||_F = \mathcal{O}_p(T^{-1/2} + g_{max,T})$.

When the hypotheses of Theorem 1 are fulfilled, a proper choice of the regularization parameters λ_1, λ_2, λ_3 ensures the existence of a local maximizer of (3), which converges at speed $\sqrt{T}$. If $\lambda_{1,T}, \lambda_{2,T}, \lambda_{3,T} \to 0$, then the estimator is root-T consistent.

Next, we show that the estimator possesses the sparsity property and hence is capable of selecting the model parameters in a sparse system. Denote by $CL(\cdot)$ the constrained likelihood. In what follows, assume without loss of generality that the true parameter matrix $\mathcal{A}$ contains p lag matrices. Each matrix has a submatrix of dimension $d_1 \times d_1$ whose elements are different from 0 in the upper left corner. The remaining elements are equal to 0. Let $\boldsymbol{A}_{k;d_1 d_1}$ indicate the respective submatrix for all k and $\boldsymbol{A}_{k;-d_1-d_1}$ the remaining elements of the respective matrix. We denote by $\boldsymbol{A}_{\cdot;d_1 d_1}$ the combined parameter matrices

$\boldsymbol{A}_{k;d_1d_1}$ over all k, and let $\boldsymbol{A}_{\cdot;-d_1-d_1}$ denote the respective combined parameter matrices $\boldsymbol{A}_{k;-d_1-d_1}$ over all k.

Lemma 1 *Assume that the assumptions for model (2) hold. If* $\lambda_{1,T}, \lambda_{2,T}, \lambda_{3,T} \to 0$ *and* $\sqrt{T}\lambda_{1,T}, \sqrt{T}\lambda_{2,T}, \sqrt{T}\lambda_{3,T} \to \infty$ *as* $T \to \infty$*, then with probability tending to 1, for any given* $\boldsymbol{A}_{k;d_1d_1}$ *satisfying* $||\boldsymbol{A}_{k;d_1d_1} - \boldsymbol{A}_{k;d_1d_1;0}||_F = O_p(T^{-1/2})$ *and any constant* Q*,*

$$CL(\boldsymbol{A}_{k;d_1d_1}, 0) = \max_{||\boldsymbol{A}_{k;-d_1-d_1}||_F \leq QT^{-1/2}} CL(\boldsymbol{A}_{k;d_1d_1}, \boldsymbol{A}_{k;-d_1-d_1}),$$

hence

$$P(\boldsymbol{A}_{k;-d_1-d_1} = 0) \to 1.$$

Finally, we show that the estimator possesses the oracle property, i.e., it chooses the true model as if it were a theoretical estimator that knows the true model structure.

We define

$$F = \left[p''_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{1;11}), \cdots, p''_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{p;d_1d_1})\right]$$

as a $pd_1 \times pd_1$ symmetric matrix containing the second derivatives of the penalty function and

$$G = \left[p'_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(A_{1;11})sgn(A_{1;11}), \cdots, p'_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(A_{p;d_1d_1})sgn(A_{p;d_1d_1})\right]$$

as a $d_1 \times pd_1$ matrix containing the first derivatives of the penalty function.

Theorem 2 *Assume that the assumptions for model (2) hold. If* $\lambda_{1,T}, \lambda_{2,T}, \lambda_{3,T} \to 0$ *and* $\sqrt{T}\lambda_{1,T}, \sqrt{T}\lambda_{2,T}, \sqrt{T}\lambda_{3,T} \to \infty$ *as* $T \to \infty$*, then with probability tending to 1, the root-T consistent local maximizer* $\boldsymbol{A} = [A_{\cdot;d_1d_1}, A_{\cdot;-d_1-d_1}]$ *from Theorem 1 must satisfy*

1. *Sparsity:* $A_{\cdot;-d_1-d_1} = \mathbf{0}$
2. *Asymptotic normality:*

$$\sqrt{T}((\boldsymbol{A}_{\cdot;d_1d_1} - \mathcal{A}_{\cdot;d_1d_1})(I(\mathcal{A}_{\cdot;d_1d_1}) + F) + G) \xrightarrow{d} N(0, I(\mathcal{A}_{\cdot;d_1d_1})) \tag{6}$$

in distribution, where $I(\mathcal{A}_{\cdot;d_1d_1})$ *is the Fisher Information knowing that* $\mathcal{A}_{\cdot;-d_1-d_1} = \mathbf{0}$*.*

III Proof - Appendix

In this section we prove the consistency and oracle property of the estimator. We prove the theorems under the assumptions made for model (2), stated in Appendix II. The proofs

follow Fan and Li (2001), Song and Bickel (2011) and Wang et al. (2007).

We further define $O_M(\cdot)$ as big O notation for elementwise convergence within a matrix and $O_V(\cdot)$ as big O notation for elementwise convergence within a vector. Likewise we define $o_M(\cdot)$ and $o_V(\cdot)$ as small o notation for matrices and vectors. Let $vec(\cdot)$ denote the vectorizing operator to convert a matrix to a vector. Further, we denote the Fisher information matrix by $I(\cdot)$, which is assumed to be finite and positive definite.

III.1 Proof of Theorem 1

Denote by $CL(\cdot)$ the constrained likelihood and by $L(\cdot)$ the likelihood. Define $CL(\boldsymbol{A}) = L(\boldsymbol{A}) - T\sum_{k=1}^{p}\sum_{i=1}^{d}\sum_{j=1}^{d} p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\boldsymbol{A}_{k;ij})$. Further define $g_T = T^{-1/2} + g_{max,T}$ and $\boldsymbol{U}$ coordinates around $\mathcal{A}$. For a large constant Q, it holds that $\{\mathcal{A} + g_T\boldsymbol{U} : ||\boldsymbol{U}||_F \leq Q\}$ is the ball around $\mathcal{A}$ and we intend to show that a local maximum with maximizer $\widehat{\boldsymbol{A}}$ lies in the ball. So we intend to show that on the surface of the ball, $||\boldsymbol{U}||_F = Q$, for any $\epsilon > 0$, there exists a large constant Q such that

$$P\{\sup_{||\boldsymbol{U}||_F=Q} CL(\mathcal{A} + g_T\boldsymbol{U}) < CL(\mathcal{A})\} \geq 1 - \epsilon. \tag{7}$$

The difference between the two penalized likelihoods $CL(\mathcal{A} + g_T\boldsymbol{U})$ and $CL(\mathcal{A})$ can be bounded from above by the likelihood and the penalization on $\widehat{\boldsymbol{A}}$ only for the pd_1^2 parameters different from 0. For the construction of the upper bound, we make use of the property $p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(0) = 0$, which holds for the $p(d-d_1)^2$ parameters which are 0. In case no parameter in $\mathcal{A}$ is 0, it will be equal, otherwise larger:

$$\begin{aligned} CL(\mathcal{A} + g_T\boldsymbol{U}) - CL(\mathcal{A}) \leq\ & L(\mathcal{A} + g_T\boldsymbol{U}) - L(\mathcal{A}) \\ & - T\sum_{k=1}^{p}\sum_{i=1}^{d_1}\sum_{j=1}^{d_1}\{p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij} + g_T\boldsymbol{U}_{k;ij}) - p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})\}. \end{aligned} \tag{8}$$

Approximating by a Taylor expansion for $\mathcal{A} + g_T\boldsymbol{U}$ around $\mathcal{A}$ gives

$$\begin{aligned} L(\mathcal{A} + g_T\boldsymbol{U}) = & L(\mathcal{A}) + (\mathcal{A} + g_T\boldsymbol{U} - \mathcal{A})L^{'}(\mathcal{A}) \\ & + \frac{1}{2}L^{''}(\mathcal{A})(\mathcal{A} + g_T\boldsymbol{U} - \mathcal{A})^{\top}(\mathcal{A} + g_T\boldsymbol{U} - \mathcal{A}) \\ & + o_p\{\frac{1}{2}L^{''}(\mathcal{A})(\mathcal{A} + g_T\boldsymbol{U} - \mathcal{A})^{\top}(\mathcal{A} + g_T\boldsymbol{U} - \mathcal{A})\} \end{aligned} \tag{9}$$

which leads to

$$L(\mathcal{A}+g_T\boldsymbol{U})-L(\mathcal{A})=g_T L^{'}(\mathcal{A})\mathrm{vec}(\boldsymbol{U})+\frac{1}{2}g_T^2\mathrm{vec}(\boldsymbol{U})^\top L^{''}(\mathcal{A})\mathrm{vec}(\boldsymbol{U}) \tag{10}$$
$$+\frac{1}{2}g_T^2\mathrm{vec}(\boldsymbol{U})^\top L^{''}(\mathcal{A})\mathrm{vec}(\boldsymbol{U})o_p\{1\}$$

Also,

$$T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij}+g_T\boldsymbol{U}_{k;ij})=T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij}) \tag{11}$$
$$+T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}(\mathcal{A}_{k;ij}+g_T\boldsymbol{U}_{k;ij}-\mathcal{A}_{k;ij})$$
$$p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})^{'}sgn(\mathcal{A}_{k;ij})$$
$$+T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}(\mathcal{A}_{k;ij}+g_T\boldsymbol{U}_{k;ij}-\mathcal{A}_{k;ij})^2$$
$$p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})^{''}$$
$$+T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}o_p(\mathcal{A}_{k;ij}+g_T\boldsymbol{U}_{k;ij}-\mathcal{A}_{k;ij})^2$$
$$p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})^{''}$$

which leads to

$$T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij}+g_T\boldsymbol{U}_{k;ij})-T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij}) \tag{12}$$
$$=T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}(g_T\boldsymbol{U}_{k;ij})p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})^{'}sgn(\mathcal{A}_{k;ij}) \tag{13}$$
$$+T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}(g_T\boldsymbol{U}_{k;ij})^2p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})^{''}$$
$$+T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}g_T^2\boldsymbol{U}_{k;ij}^2o_p(1)^2p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})^{''}$$

Recall that $L^{''}(\mathcal{A})=-TI(\mathcal{A})$.

Hence,

$$
\begin{aligned}
CL(\mathcal{A}+g_T\boldsymbol{U})-CL(\mathcal{A}) \leq & bL^{'}(\mathcal{A})^{\top}\mathrm{vec}(\boldsymbol{U}) \\
& -\frac{1}{2}Tg_T^2\mathrm{vec}(\boldsymbol{U})^{\top}I(\mathcal{A})\mathrm{vec}(\boldsymbol{U})(1+o_p(1)) \\
& -T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1} g_T p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})^{'} sgn(\mathcal{A}_{k;ij})\boldsymbol{U}_{k;ij} \\
& -T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1} (g_T\boldsymbol{U}_{k;ij})^2 p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})^{''}(1+o_p(1))
\end{aligned} \tag{14}
$$

If the right-hand side of the inequality is smaller 0, the inequality holds. Note that it holds $T^{-1/2}L(\mathcal{A})^{'} = \mathcal{O}_V(1)$. It follows that the first term on the right-hand side is of order $\mathcal{O}_V(T^{1/2}g_T)$. The second term is of order $\mathcal{O}_p(Tg_T^2)$, and it holds $\mathcal{O}_p(T^{1/2}g_T) = \mathcal{O}_p(Tg_T^2)$. For a sufficiently large Q, the second term dominates the first term uniformly in $||\boldsymbol{U}||_F = Q$. The third and fourth term are bounded by

$$
Tg_T\sum_{k=1}^{p}\sum_{i,j=1}^{d_1} g_{max,T}\boldsymbol{U}_{k;ij} + Tg_T^2\sum_{k=1}^{p}\sum_{i,j=1}^{d_1}\boldsymbol{U}_{k;ij}^2 \max\left(\frac{\partial^2 p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}}{\partial A_{k;ij}^2} : A_{k;ij}\neq 0\right)(1+o_p(1)), \tag{15}
$$

and therefore are $\mathcal{O}_p(Tg_T)$ and $\mathcal{O}_p(Tg_T^2)$. Since $\max(\frac{\partial^2 p_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}}{\partial A_{k;ij}^2} : A_{k;ij}\neq 0) \to 0$ and the term is of $\mathcal{O}_p(Tg_T^2)$, it is dominated by the second term in case of a large Q. Also the third term is dominated by the second term since it is of order $\mathcal{O}_p(Tg_Tg_{max,T})$ which is dominated by $\mathcal{O}_p(Tg_T^2)$ and Q takes on a larger effect in the second term. Therefore, the negativity of the second term ensures the right-hand side to be smaller 0 in case of a large Q. Hence (7) holds. This implies that there exists a local maximizer $\widehat{\boldsymbol{A}}$ for which $||\widehat{\boldsymbol{A}}-\mathcal{A}||_F = \mathcal{O}_p(g_T)$. This completes the proof of the theorem.

III.2 Proof of Lemma 1

We carry out the proof by showing that all parameters in $\boldsymbol{A}_{k;-d_1-d_1}$ for all k cannot be different from 0 since this would be a contradiction. One has

$$
\frac{\partial CL(\widehat{\boldsymbol{A}}_k)}{\partial A_{k;ij}} = \frac{\partial L(\widehat{\boldsymbol{A}}_k)}{\partial A_{k;ij}} - Tp^{'}_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\widehat{A}_{k;ij})sgn(\widehat{A}_{k;ij}), \tag{16}
$$

hence for a consistent selection of $\boldsymbol{A}_{k;-d_1-d_1}$ all parameters have to be 0. Otherwise the first derivative of the constrained likelihood would not equal the unconstrained one, which is 0.

It is sufficient to show that $\frac{\partial CL(\boldsymbol{A}_k)}{\partial A_{k;ij}} \neq 0$ if and only if $A_{k;ij} \neq 0$. Hence we will show that with probability tending to 1 for $T \to \infty$, for any $\boldsymbol{A}_{k;d_1d_1}$ satisfying $\boldsymbol{A}_{k;d_1d_1} - \mathcal{A}_{k;d_1d_1} = O_M(T^{-1/2})$ and for some small $\epsilon_T = QT^{-1/2}$ and $i, j = 1, \ldots, d_1$,

$$\frac{\partial CL(\boldsymbol{A}_k)}{\partial A_{k;ij}} < 0 \quad \text{for } 0 < A_{k;ij} < \epsilon_T \tag{17}$$

$$> 0 \quad \text{for } -\epsilon_T < A_{k;ij} < 0 \tag{18}$$

By Taylor's expansion,

$$\frac{\partial CL(\widehat{\boldsymbol{A}}_k)}{\partial A_{k;ij}} = \frac{\partial L(\widehat{\boldsymbol{A}}_k)}{\partial A_{k;ij}} - Tp'_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\widehat{A}_{k;ij})sgn(\widehat{A}_{k;ij}) \tag{19}$$

$$= \frac{\partial L(\mathcal{A}_k)}{\partial A_{k;ij}} + \sum_{l_1=1}^{d_2}\sum_{l_2=1}^{d_2} \frac{\partial^2 L(\mathcal{A}_k)}{\partial A_{k;ij}\partial A_{k;l_1l_2}}(\widehat{A}_{k;l_1l_2} - \mathcal{A}_{k;l_1l_2}) \tag{20}$$

$$+ \sum_{l_1=1}^{d_2}\sum_{l_2=1}^{d_2}\sum_{l_3=1}^{d_2}\sum_{l_4=1}^{d_2} \frac{\partial^3 L(\boldsymbol{A}_k^*)}{\partial A_{k;ij}\partial A_{k;l_1l_2}\partial A_{k;l_3l_4}} \times (\widehat{A}_{k;l_1l_2} - \mathcal{A}_{k;l_1l_2})(\widehat{A}_{k;l_3l_4} - \mathcal{A}_{k;l_3l_4})$$

$$- Tp'_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\widehat{A}_{k;ij})sgn(\widehat{A}_{k;ij})$$

with $\boldsymbol{A}_k^*$ lying between $\widehat{\boldsymbol{A}}_k$ and $\mathcal{A}_k$.

Recall that

$$T^{-1}\frac{\partial L(\mathcal{A}_k)}{\partial A_{k;ij}} = O(T^{-1/2})$$

$$T^{-1}\frac{\partial^2 L(\mathcal{A}_k)}{\partial A_{k;ij}\partial A_{k;l_1l_2}} = \mathrm{E}\left(\frac{\partial^2 L(\mathcal{A}_k)}{\partial A_{k;ij}\partial A_{k;l_1l_2}}\right) + o(1)$$

The first term is therefore of order $O(T^{1/2})$. The second term is also of order $O(T^{1/2})$ because it consists of the Fisher information matrix and $o(1)$, where the latter is negligible because o goes to 0 faster than O. The third term is obviously faster at 0 due to the squared O_M, meaning it is bounded by $O_P(T^{-1/2})^2$, hence it goes faster to 0 than the first and second term. It follows that

$$\frac{\partial CL(\boldsymbol{A}_k)}{\partial A_{k;ij}} = -Tp'_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(A_{k;ij})sgn(A_{k;ij}) + O(T^{1/2}) \tag{21}$$

The first term dominates, because $\sqrt{T}g_{max,T} \to \infty$. Hence the sign of $A_{k;ij}$ determines the sign of $\frac{\partial CL(\boldsymbol{A}_k)}{\partial A_{k;ij}}$. Hence the inequalities (17) and (18) hold, which implies that $\frac{\partial CL(\boldsymbol{A}_k)}{\partial A_{k;ij}}$ can only be 0 if and only if $A_{k;ij} = 0$. This completes the proof.

III.3 Proof of Theorem 2

From Lemma 1 there follows 1. It can be easily shown that there exists an $\widehat{\boldsymbol{A}}_{\cdot;d_1d_1}$ in Theorem 1 that is a root-T consistent local maximizer of $CL((\boldsymbol{A}_{\cdot;d_1d_1}, \boldsymbol{0}))$ that satisfies the likelihood equations

$$\frac{\partial CL(\boldsymbol{A})}{\partial A_{k;ij}}\bigg|_{\boldsymbol{A}=[\boldsymbol{A}_{\cdot;d_1d_1}, \boldsymbol{A}_{\cdot;-d_1-d_1}]} = 0 \qquad \text{for } i = 1, \cdots, d_1; j = 1, \cdots, d_1 \tag{22}$$

Recall that $\boldsymbol{A}_{\cdot;d_1d_1}$ is a consistent estimator,

$$\begin{aligned}
&\frac{\partial L(\boldsymbol{A})}{\partial A_{k;ij}}\bigg|_{\boldsymbol{A}=[\boldsymbol{A}_{\cdot;d_1d_1}, \boldsymbol{A}_{\cdot;-d_1-d_1}]} - Tp^{'}_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(A_{k;ij})sgn(A_{k;ij}) && (23)\\
&= \frac{\partial L(\mathcal{A})}{\partial A_{k;ij}} + \sum_{l_1=1}^{p}\sum_{l_2=1}^{d_1}\sum_{l_3=1}^{d_1}\left(\frac{\partial^2 L(\mathcal{A})}{\partial A_{k;ij}\partial A_{l_1;l_2l_3}} + o_P(1)\right)(A_{k;ij} - \mathcal{A}_{k;ij}) && (24)\\
&- T\Big(p^{'}_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})sgn(\mathcal{A}_{k;ij})\\
&+(p^{''}_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij}) + o_P(1))(A_{k;ij} - \mathcal{A}_{k;ij})\Big).
\end{aligned}$$

Setting the first derivative equal to 0 and rearranging terms gives

$$\begin{aligned}
(A_{k;ij} - \mathcal{A}_{k;ij}) &= -\frac{\frac{\partial L(\mathcal{A})}{\partial A_{k;ij}} - Tp^{'}_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})sgn(\mathcal{A}_{k;ij})}{H - TK}\\
&= -\frac{\frac{1}{T}\frac{\partial L(\mathcal{A})}{\partial A_{k;ij}} - p^{'}_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})sgn(\mathcal{A}_{k;ij})}{\frac{1}{T}H - K},
\end{aligned}$$

whereas $H = \sum_{l_1=1}^{p}\sum_{l_2=1}^{d_1}\sum_{l_3=1}^{d_1}\left(\frac{\partial^2 L(\mathcal{A})}{\partial A_{k;ij}\partial A_{l_1;l_2l_3}} + o_P(1)\right)$ and $K = p^{''}_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij}) + o_P(1)$.

The nominator converges in distribution by the Central Limit Theorem to

$$\frac{1}{T}\frac{\partial L(\mathcal{A})}{\partial A_{k;ij}} - p^{'}_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})sgn(\mathcal{A}_{k;ij}) \xrightarrow{d} N(0, \frac{I(\mathcal{A}_{\cdot;d_1d_1})_{k;ij}}{T}) - G_{k;ij} \tag{25}$$

By Slutsky's Theorem, the denominator goes to

$$
\begin{aligned}
&-\frac{1}{T}\sum_{l_1=1}^{p}\sum_{l_2=1}^{d_1}\sum_{l_3=1}^{d_1}\left(\frac{\partial^2 L(\mathcal{A})}{\partial A_{k;ij}\partial A_{l_1;l_2l_3}}+o_P(1)\right)\\
&+(p''_{\lambda_{1,T},\lambda_{2,T},\lambda_{3,T}}(\mathcal{A}_{k;ij})+o_P(1))\to I(\mathcal{A}_{\cdot;d_1d_1})_{k;ij}+F_{k;ij}
\end{aligned}
\tag{26}
$$

Combining the two results and writing this in matrix form gives

$$
\begin{aligned}
(\boldsymbol{A}_{\cdot;d_1d_1}-\mathcal{A}_{\cdot;d_1d_1})&\xrightarrow{d}N(0,\frac{I(\mathcal{A}_{\cdot;d_1d_1})}{T}(I(\mathcal{A}_{\cdot;d_1d_1})+F)^{-2})-G(I(\mathcal{A}_{\cdot;d_1d_1})+F)^{-1}\\
(\boldsymbol{A}_{\cdot;d_1d_1}-\mathcal{A}_{\cdot;d_1d_1})+G(I(\mathcal{A}_{\cdot;d_1d_1})+F)^{-1}&\xrightarrow{d}N(0,\frac{I(\mathcal{A}_{\cdot;d_1d_1})}{T}(I(\mathcal{A}_{\cdot;d_1d_1})+F)^{-2})\\
\sqrt{T}((\boldsymbol{A}_{\cdot;d_1d_1}-\mathcal{A}_{\cdot;d_1d_1})(I(\mathcal{A}_{\cdot;d_1d_1})+F)+G)&\xrightarrow{d}N(0,I(\mathcal{A}_{\cdot;d_1d_1}))
\end{aligned}
$$

Hence by applying Slutsky's Theorem and the Central Limit Theorem, we find

$$
\sqrt{T}((\boldsymbol{A}_{\cdot;d_1d_1}-\mathcal{A}_{\cdot;d_1d_1})(I(\mathcal{A}_{\cdot;d_1d_1})+F)+G)\xrightarrow{d}N(0,I(\mathcal{A}_{\cdot;d_1d_1}))
\tag{27}
$$

This completes the proof.

IV Tables - Appendix

Table 4: Simulation $d = 50$ with 3 lags, $T = 100$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**3.04**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**0.33**
	TLASSO	0.00	0.00	0.00	0.24	0.01	0.04	0.05	0.87	0.87	1.27
	SCAD	0.00	0.65	0.00	0.90	0.18	0.22	0.04	0.85	0.87	0.89
	LASSO	0.00	0.67	0.00	0.95	0.15	0.38	0.05	0.87	0.88	0.79
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.01**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**3.24**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.01**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**0.44**
	TLASSO	0.01	0.50	0.01	0.96	0.27	0.50	0.06	0.88	0.89	1.27
	SCAD	0.00	0.67	0.00	0.98	0.05	0.63	0.06	0.81	0.87	0.91
	LASSO	0.00	0.67	0.00	0.99	0.03	0.75	0.05	0.84	0.86	0.81
M1	TriSNAR$_G$	0.00	0.23	0.78	0.24	0.65	0.06	0.05	0.84	0.84	3.51
	TriSNAR$_A$	0.00	0.22	0.72	0.22	0.65	0.05	0.05	0.84	0.84	0.46
	TLASSO	**0.00**	**0.00**	**0.22**	**0.21**	**0.65**	**0.03**	**0.07**	**0.87**	**0.87**	**1.35**
	SCAD	0.00	0.66	0.20	0.84	0.66	0.37	0.06	0.83	0.86	0.95
	LASSO	0.00	0.67	0.13	0.87	0.63	0.43	0.07	0.86	0.87	0.84
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.49**	**0.01**	**0.48**	**0.09**	**0.05**	**0.82**	**0.83**	**3.32**
	TriSNAR$_A$	**0.00**	**0.00**	**0.49**	**0.01**	**0.48**	**0.09**	**0.05**	**0.82**	**0.83**	**0.50**
	TLASSO	0.00	0.50	0.73	0.94	0.71	0.44	0.07	0.87	0.88	1.22
	SCAD	0.00	0.67	0.02	0.93	0.51	0.62	0.08	0.78	0.86	0.82
	LASSO	0.00	0.67	0.01	0.94	0.50	0.64	0.07	0.83	0.85	0.73
M1/M3	TriSNAR$_G$	**0.30**	**0.09**	**0.61**	**0.45**	**0.85**	**0.35**	**0.07**	**0.88**	**0.89**	**2.50**
	TriSNAR$_A$	0.31	0.09	0.61	0.51	0.86	0.38	0.07	0.88	0.89	0.34
	TLASSO	0.50	0.00	0.58	0.69	0.87	0.36	0.07	0.89	0.89	1.15
	SCAD	0.00	0.33	0.11	0.88	0.85	0.72	0.09	0.83	0.91	0.81
	LASSO	0.00	0.33	0.06	0.90	0.81	0.74	0.08	0.87	0.89	0.71
NS1	TriSNAR$_G$	0.00	0.04	0.98	0.03	0.98	0.01	0.06	0.84	0.84	3.34
	TriSNAR$_A$	0.00	0.20	0.88	0.05	0.98	0.02	0.06	0.83	0.84	0.40
	TLASSO	**0.00**	**0.00**	**0.49**	**0.00**	**0.97**	**0.00**	**0.07**	**0.87**	**0.88**	**1.38**
	SCAD	0.00	0.67	0.56	0.49	0.97	0.31	0.08	0.82	0.86	0.83
	LASSO	0.00	0.67	0.47	0.52	0.97	0.34	0.07	0.86	0.88	0.73

Table 5: Simulation $d = 50$ with 3 lags, $T = 200$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**2.70**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**0.33**
	TLASSO	0.00	0.00	0.00	0.93	0.00	0.28	0.04	0.85	0.85	1.57
	SCAD	0.00	0.06	0.00	0.10	0.02	0.00	0.03	0.85	0.85	1.64
	LASSO	0.00	0.38	0.00	0.53	0.02	0.04	0.05	0.87	0.87	1.36
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**3.06**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**0.31**
	TLASSO	0.00	0.50	0.00	0.93	0.00	0.29	0.05	0.86	0.86	1.63
	SCAD	0.00	0.67	0.00	0.98	0.00	0.56	0.04	0.83	0.85	1.67
	LASSO	0.00	0.67	0.00	0.99	0.00	0.71	0.04	0.84	0.85	1.40
M1	TriSNAR$_G$	0.00	0.00	0.80	0.00	0.66	0.00	0.05	0.84	0.84	3.39
	TriSNAR$_A$	0.00	0.00	0.79	0.00	0.66	0.00	0.05	0.84	0.84	0.50
	TLASSO	**0.00**	**0.00**	**0.10**	**0.22**	**0.48**	**0.05**	**0.06**	**0.85**	**0.86**	**1.98**
	SCAD	0.00	0.54	0.28	0.48	0.62	0.10	0.05	0.83	0.84	1.73
	LASSO	0.00	0.62	0.16	0.56	0.60	0.11	0.07	0.86	0.87	1.45
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.80**	**0.00**	**0.66**	**0.00**	**0.05**	**0.82**	**0.83**	**3.22**
	TriSNAR$_A$	**0.00**	**0.00**	**0.80**	**0.00**	**0.66**	**0.00**	**0.05**	**0.82**	**0.83**	**0.42**
	TLASSO	0.00	0.50	0.44	0.81	0.63	0.24	0.06	0.85	0.85	1.89
	SCAD	0.00	0.67	0.00	0.93	0.32	0.56	0.07	0.80	0.83	1.51
	LASSO	0.00	0.67	0.00	0.93	0.30	0.51	0.06	0.82	0.83	1.26
M1/M3	TriSNAR$_G$	0.34	0.10	0.66	0.24	0.80	0.13	0.07	0.87	0.88	2.36
	TriSNAR$_A$	**0.34**	**0.10**	**0.67**	**0.22**	**0.81**	**0.11**	**0.07**	**0.86**	**0.87**	**0.31**
	TLASSO	0.50	0.00	0.62	0.46	0.84	0.13	0.07	0.89	0.89	1.64
	SCAD	0.00	0.33	0.09	0.81	0.74	0.43	0.08	0.85	0.88	1.45
	LASSO	0.00	0.33	0.04	0.85	0.68	0.50	0.07	0.87	0.89	1.20
NS1	TriSNAR$_G$	0.00	0.00	0.98	0.00	0.98	0.00	0.06	0.84	0.84	3.36
	TriSNAR$_A$	0.00	0.09	0.92	0.03	0.98	0.01	0.06	0.84	0.84	0.47
	TLASSO	**0.00**	**0.00**	**0.00**	**0.00**	**0.91**	**0.00**	**0.06**	**0.82**	**0.83**	**1.74**
	SCAD	0.00	0.66	0.53	0.32	0.97	0.17	0.07	0.82	0.84	1.49
	LASSO	0.00	0.67	0.40	0.36	0.96	0.21	0.07	0.86	0.87	1.25

Table 6: Simulation $d = 50$ with 3 lags, $T = 500$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**0.64**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**0.22**
	TLASSO	0.00	0.00	0.00	0.13	0.00	0.01	0.04	0.85	0.85	2.75
	SCAD	0.00	0.16	0.00	0.22	0.00	0.03	0.02	0.85	0.85	1.36
	LASSO	0.00	0.67	0.00	0.95	0.00	0.34	0.04	0.85	0.85	1.21
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**0.71**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**0.28**
	TLASSO	0.00	0.49	0.00	0.85	0.00	0.19	0.04	0.85	0.85	2.82
	SCAD	0.00	0.66	0.00	0.97	0.00	0.54	0.02	0.83	0.84	1.31
	LASSO	0.00	0.66	0.00	0.98	0.00	0.65	0.03	0.84	0.84	1.17
M1	TriSNAR$_G$	0.00	0.10	0.79	0.09	0.65	0.01	0.05	0.84	0.84	1.55
	TriSNAR$_A$	0.00	0.14	0.72	0.14	0.61	0.02	0.05	0.84	0.84	0.32
	TLASSO	**0.00**	**0.00**	**0.00**	**0.42**	**0.02**	**0.05**	**0.05**	**0.83**	**0.83**	**3.90**
	SCAD	0.00	0.32	0.40	0.19	0.64	0.02	0.05	0.84	0.84	2.04
	LASSO	0.00	0.67	0.00	0.89	0.10	0.35	0.05	0.83	0.84	1.86
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.02**	**0.00**	**0.02**	**0.10**	**0.03**	**0.81**	**0.81**	**1.36**
	TriSNAR$_A$	**0.00**	**0.00**	**0.02**	**0.00**	**0.02**	**0.10**	**0.03**	**0.81**	**0.81**	**0.34**
	TLASSO	0.00	0.50	0.00	0.60	0.25	0.16	0.05	0.83	0.83	3.95
	SCAD	0.00	0.67	0.00	0.94	0.04	0.54	0.05	0.80	0.82	1.75
	LASSO	0.00	0.67	0.00	0.91	0.04	0.38	0.05	0.82	0.82	1.60
M1/M3	TriSNAR$_G$	0.00	0.17	0.42	0.18	0.46	0.13	0.06	0.85	0.85	0.53
	TriSNAR$_A$	**0.00**	**0.14**	**0.30**	**0.15**	**0.49**	**0.09**	**0.06**	**0.85**	**0.85**	**0.24**
	TLASSO	0.50	0.00	0.51	0.26	0.71	0.03	0.07	0.88	0.88	3.43
	SCAD	0.00	0.31	0.06	0.41	0.61	0.07	0.07	0.85	0.86	1.61
	LASSO	0.00	0.33	0.01	0.62	0.52	0.13	0.07	0.88	0.88	1.45
NS1	TriSNAR$_G$	0.00	0.64	0.04	0.44	0.93	0.25	0.06	0.82	0.83	1.52
	TriSNAR$_A$	0.00	0.67	0.02	0.35	0.93	0.17	0.06	0.81	0.82	0.30
	TLASSO	**0.00**	**0.10**	**0.00**	**0.00**	**0.87**	**0.00**	**0.05**	**0.81**	**0.82**	**3.30**
	SCAD	0.00	0.65	0.09	0.43	0.93	0.27	0.06	0.81	0.82	1.77
	LASSO	0.00	0.67	0.00	0.54	0.92	0.35	0.06	0.82	0.83	1.61

Table 7: Simulation $d = 50$ with 3 lags, $T = 1000$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**0.61**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**0.42**
	TLASSO	0.00	0.00	0.00	0.93	0.00	0.24	0.03	0.85	0.85	4.72
	SCAD	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**2.43**
	LASSO	0.00	0.49	0.00	0.54	0.00	0.05	0.04	0.85	0.85	2.26
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.01**	**0.84**	**0.84**	**0.61**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.01**	**0.84**	**0.84**	**0.41**
	TLASSO	0.00	0.49	0.00	0.81	0.00	0.15	0.03	0.84	0.84	5.15
	SCAD	0.00	0.66	0.00	0.97	0.00	0.54	0.02	0.84	0.84	2.18
	LASSO	0.00	0.67	0.00	0.98	0.00	0.62	0.03	0.84	0.84	2.00
M1	TriSNAR$_G$	**0.00**	**0.04**	**0.00**	**0.01**	**0.00**	**0.13**	**0.03**	**0.82**	**0.82**	**1.76**
	TriSNAR$_A$	0.00	0.04	0.00	0.01	0.00	0.13	0.03	0.82	0.82	0.60
	TLASSO	0.00	0.00	0.00	0.87	0.00	0.31	0.04	0.82	0.83	7.39
	SCAD	0.00	0.67	0.00	0.78	0.04	0.32	0.04	0.82	0.83	4.23
	LASSO	0.00	0.66	0.00	0.67	0.03	0.15	0.05	0.83	0.84	4.06
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.05**	**0.04**	**0.04**	**0.09**	**0.03**	**0.81**	**0.81**	**1.68**
	TriSNAR$_A$	**0.00**	**0.00**	**0.05**	**0.04**	**0.04**	**0.09**	**0.03**	**0.81**	**0.81**	**0.59**
	TLASSO	0.00	0.50	0.00	0.83	0.00	0.40	0.04	0.81	0.81	8.43
	SCAD	0.00	0.67	0.00	0.94	0.00	0.58	0.05	0.80	0.81	3.48
	LASSO	0.00	0.66	0.00	0.90	0.00	0.34	0.04	0.81	0.81	3.31
M1/M3	TriSNAR$_G$	0.00	0.00	0.80	0.00	0.66	0.00	0.06	0.85	0.86	0.85
	TriSNAR$_A$	0.00	0.00	0.79	0.01	0.66	0.00	0.06	0.85	0.86	0.63
	TLASSO	0.40	0.00	0.48	0.20	0.60	0.08	0.07	0.87	0.87	7.30
	SCAD	**0.00**	**0.19**	**0.07**	**0.09**	**0.55**	**0.01**	**0.06**	**0.85**	**0.86**	**3.09**
	LASSO	0.00	0.29	0.00	0.34	0.37	0.08	0.06	0.87	0.87	2.87
NS1	TriSNAR$_G$	0.00	0.54	0.00	0.06	0.89	0.02	0.04	0.81	0.81	1.27
	TriSNAR$_A$	0.00	0.58	0.00	0.12	0.93	0.04	0.05	0.81	0.81	0.54
	TLASSO	**0.00**	**0.02**	**0.00**	**0.00**	**0.87**	**0.00**	**0.04**	**0.81**	**0.81**	**6.08**
	SCAD	0.00	0.67	0.00	0.46	0.92	0.25	0.05	0.81	0.81	3.48
	LASSO	0.00	0.67	0.00	0.49	0.92	0.31	0.06	0.82	0.82	3.31

Table 8: Simulation $d = 10$ with 3 lags, $T = 100$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.00**
	TLASSO	0.00	0.00	0.00	0.50	0.00	0.15	0.09	0.86	0.86	0.02
	SCAD	0.00	0.29	0.00	0.40	0.07	0.11	0.08	0.85	0.86	0.04
	LASSO	0.00	0.51	0.00	0.65	0.07	0.22	0.11	0.87	0.87	0.03
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.84**	**0.84**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.84**	**0.84**	**0.01**
	TLASSO	0.00	0.50	0.00	0.89	0.00	0.59	0.10	0.85	0.85	0.02
	SCAD	0.00	0.65	0.00	0.89	0.02	0.56	0.09	0.83	0.85	0.04
	LASSO	0.00	0.66	0.00	0.93	0.00	0.68	0.11	0.84	0.85	0.03
M1	TriSNAR$_G$	0.00	0.00	0.80	0.00	0.66	0.00	0.11	0.84	0.84	0.01
	TriSNAR$_A$	0.00	0.01	0.80	0.01	0.65	0.00	0.11	0.84	0.84	0.01
	TLASSO	**0.00**	**0.02**	**0.03**	**0.21**	**0.27**	**0.11**	**0.12**	**0.84**	**0.84**	**0.03**
	SCAD	0.00	0.36	0.52	0.30	0.63	0.11	0.12	0.84	0.85	0.05
	LASSO	0.00	0.60	0.30	0.51	0.55	0.26	0.15	0.86	0.87	0.04
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.80**	**0.00**	**0.65**	**0.00**	**0.11**	**0.83**	**0.83**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.80**	**0.00**	**0.65**	**0.00**	**0.11**	**0.83**	**0.83**	**0.01**
	TLASSO	0.00	0.50	0.26	0.65	0.47	0.41	0.13	0.84	0.85	0.03
	SCAD	**0.00**	**0.63**	**0.14**	**0.58**	**0.43**	**0.34**	**0.13**	**0.81**	**0.84**	**0.04**
	LASSO	0.00	0.66	0.07	0.69	0.34	0.45	0.13	0.82	0.84	0.04
M1/M3	TriSNAR$_G$	0.26	0.15	0.57	0.24	0.78	0.19	0.16	0.87	0.89	0.01
	TriSNAR$_A$	**0.26**	**0.15**	**0.62**	**0.21**	**0.81**	**0.17**	**0.16**	**0.87**	**0.89**	**0.01**
	TLASSO	**0.50**	**0.00**	**0.72**	**0.12**	**0.87**	**0.08**	**0.16**	**0.89**	**0.89**	**0.02**
	SCAD	0.00	0.33	0.28	0.48	0.75	0.39	0.18	0.85	0.89	0.04
	LASSO	0.00	0.33	0.18	0.53	0.66	0.43	0.16	0.87	0.89	0.03
NS1	TriSNAR$_G$	0.00	0.02	0.89	0.01	0.90	0.01	0.13	0.85	0.85	0.01
	TriSNAR$_A$	0.00	0.02	0.89	0.01	0.90	0.01	0.13	0.85	0.85	0.01
	TLASSO	**0.00**	**0.04**	**0.15**	**0.01**	**0.71**	**0.00**	**0.13**	**0.84**	**0.85**	**0.03**
	SCAD	0.00	0.51	0.57	0.31	0.88	0.17	0.15	0.84	0.86	0.05
	LASSO	0.00	0.64	0.38	0.44	0.83	0.28	0.16	0.86	0.87	0.04

Table 9: Simulation $d = 10$ with 3 lags, $T = 200$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.00**
	TLASSO	0.00	0.04	0.00	0.54	0.00	0.25	0.08	0.85	0.85	0.04
	SCAD	0.00	0.21	0.00	0.28	0.00	0.12	0.06	0.85	0.85	0.04
	LASSO	0.00	0.53	0.00	0.73	0.00	0.45	0.10	0.85	0.86	0.03
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.01**	**0.00**	**0.00**	**0.04**	**0.84**	**0.84**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.01**	**0.00**	**0.00**	**0.04**	**0.84**	**0.84**	**0.01**
	TLASSO	0.00	0.50	0.00	0.87	0.00	0.56	0.08	0.84	0.84	0.04
	SCAD	0.00	0.65	0.00	0.90	0.00	0.56	0.07	0.83	0.84	0.03
	LASSO	0.00	0.66	0.00	0.92	0.00	0.64	0.09	0.84	0.84	0.03
M1	TriSNAR$_G$	0.00	0.29	0.58	0.21	0.49	0.13	0.11	0.83	0.84	0.01
	TriSNAR$_A$	0.00	0.25	0.57	0.18	0.49	0.11	0.11	0.83	0.84	0.01
	TLASSO	**0.00**	**0.08**	**0.00**	**0.40**	**0.03**	**0.20**	**0.10**	**0.83**	**0.83**	**0.05**
	SCAD	0.00	0.09	0.63	0.07	0.62	0.02	0.11	0.84	0.84	0.05
	LASSO	0.00	0.54	0.21	0.48	0.38	0.26	0.13	0.85	0.85	0.04
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.77**	**0.00**	**0.63**	**0.00**	**0.10**	**0.82**	**0.82**	**0.01**
	TriSNAR$_A$	0.00	0.00	0.77	0.00	0.63	0.00	0.10	0.82	0.82	0.01
	TLASSO	0.00	0.49	0.04	0.56	0.20	0.31	0.11	0.82	0.83	0.05
	SCAD	0.00	0.62	0.03	0.54	0.25	0.28	0.11	0.81	0.82	0.04
	LASSO	0.00	0.66	0.01	0.65	0.18	0.37	0.11	0.82	0.82	0.04
M1/M3	TriSNAR$_G$	0.00	0.32	0.27	0.33	0.48	0.23	0.15	0.86	0.87	0.01
	TriSNAR$_A$	**0.00**	**0.30**	**0.28**	**0.27**	**0.60**	**0.17**	**0.16**	**0.86**	**0.87**	**0.00**
	TLASSO	0.49	0.01	0.67	0.09	0.78	0.05	0.15	0.88	0.88	0.05
	SCAD	0.00	0.31	0.26	0.30	0.66	0.17	0.16	0.86	0.87	0.04
	LASSO	0.00	0.33	0.15	0.41	0.55	0.24	0.15	0.87	0.88	0.03
NS1	TriSNAR$_G$	0.00	0.52	0.77	0.42	0.82	0.13	0.13	0.84	0.84	0.01
	TriSNAR$_A$	0.00	0.46	0.60	0.31	0.82	0.11	0.13	0.83	0.84	0.01
	TLASSO	**0.00**	**0.09**	**0.01**	**0.02**	**0.55**	**0.00**	**0.11**	**0.83**	**0.83**	**0.05**
	SCAD	0.00	0.38	0.53	0.21	0.84	0.10	0.13	0.83	0.84	0.05
	LASSO	0.00	0.61	0.33	0.34	0.80	0.20	0.15	0.86	0.86	0.04

Table 10: Simulation $d = 10$ with 3 lags, $T = 500$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.01**
	TLASSO	0.00	0.00	0.00	0.45	0.00	0.13	0.07	0.85	0.85	0.07
	SCAD	0.00	0.06	0.00	0.08	0.00	0.02	0.05	0.85	0.85	0.04
	LASSO	0.00	0.48	0.00	0.58	0.00	0.17	0.08	0.85	0.85	0.04
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.84**	**0.84**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.84**	**0.84**	**0.01**
	TLASSO	0.00	0.50	0.00	0.87	0.00	0.52	0.07	0.84	0.84	0.08
	SCAD	0.00	0.65	0.00	0.89	0.00	0.56	0.05	0.83	0.84	0.04
	LASSO	0.00	0.66	0.00	0.92	0.00	0.62	0.07	0.84	0.84	0.03
M1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.13**	**0.08**	**0.82**	**0.83**	**0.01**
	TriSNAR$_A$	0.00	0.01	0.00	0.00	0.01	0.13	0.08	0.82	0.83	0.01
	TLASSO	0.00	0.09	0.00	0.31	0.00	0.14	0.09	0.83	0.83	0.10
	SCAD	0.00	0.51	0.02	0.30	0.20	0.11	0.09	0.82	0.83	0.06
	LASSO	0.00	0.65	0.00	0.60	0.07	0.32	0.11	0.83	0.83	0.06
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.12**	**0.14**	**0.09**	**0.14**	**0.07**	**0.81**	**0.81**	**0.01**
	TriSNAR$_A$	0.00	0.01	0.12	0.14	0.09	0.14	0.07	0.81	0.81	0.01
	TLASSO	0.00	0.50	0.00	0.59	0.00	0.30	0.09	0.81	0.82	0.12
	SCAD	0.00	0.60	0.00	0.51	0.03	0.22	0.07	0.81	0.81	0.05
	LASSO	0.00	0.65	0.00	0.62	0.01	0.30	0.09	0.81	0.82	0.05
M1/M3	TriSNAR$_G$	0.06	0.03	0.66	0.05	0.56	0.04	0.14	0.86	0.86	0.01
	TriSNAR$_A$	**0.01**	**0.03**	**0.45**	**0.09**	**0.47**	**0.04**	**0.14**	**0.85**	**0.86**	**0.01**
	TLASSO	0.12	0.22	0.24	0.43	0.33	0.28	0.13	0.85	0.86	0.10
	SCAD	0.00	0.27	0.11	0.29	0.35	0.13	0.15	0.85	0.86	0.05
	LASSO	0.00	0.31	0.06	0.40	0.26	0.20	0.14	0.86	0.87	0.04
NS1	TriSNAR$_G$	0.00	0.25	0.05	0.06	0.53	0.02	0.10	0.82	0.83	0.01
	TriSNAR$_A$	0.00	0.45	0.04	0.14	0.68	0.06	0.11	0.82	0.83	0.01
	TLASSO	**0.00**	**0.10**	**0.00**	**0.03**	**0.50**	**0.01**	**0.10**	**0.82**	**0.83**	**0.10**
	SCAD	0.00	0.62	0.05	0.29	0.71	0.15	0.11	0.82	0.83	0.06
	LASSO	0.00	0.66	0.01	0.49	0.64	0.32	0.12	0.83	0.84	0.06

Table 11: Simulation $d = 10$ with 3 lags, $T = 1000$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.01**
	TLASSO	0.00	0.04	0.00	0.31	0.00	0.14	0.07	0.85	0.85	0.12
	SCAD	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.05**	**0.85**	**0.85**	**0.06**
	LASSO	0.00	0.50	0.00	0.66	0.00	0.39	0.07	0.85	0.85	0.06
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.84**	**0.84**	**0.01**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.84**	**0.84**	**0.01**
	TLASSO	0.00	0.48	0.00	0.80	0.00	0.45	0.06	0.84	0.84	0.14
	SCAD	0.00	0.65	0.00	0.89	0.00	0.54	0.04	0.84	0.84	0.05
	LASSO	0.00	0.66	0.00	0.91	0.00	0.58	0.06	0.84	0.84	0.04
M1	TriSNAR$_G$	**0.00**	**0.12**	**0.00**	**0.27**	**0.00**	**0.16**	**0.07**	**0.82**	**0.83**	**0.02**
	TriSNAR$_A$	0.00	0.26	0.00	0.33	0.00	0.15	0.07	0.82	0.82	0.01
	TLASSO	0.00	0.32	0.00	0.51	0.00	0.28	0.08	0.82	0.82	0.18
	SCAD	0.00	0.45	0.00	0.31	0.05	0.12	0.08	0.82	0.83	0.10
	LASSO	0.00	0.67	0.00	0.72	0.00	0.47	0.10	0.82	0.83	0.09
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.11**	**0.00**	**0.11**	**0.05**	**0.81**	**0.81**	**0.02**
	TriSNAR$_A$	0.00	0.01	0.00	0.12	0.00	0.11	0.05	0.81	0.81	0.02
	TLASSO	0.00	0.50	0.00	0.57	0.00	0.27	0.08	0.81	0.81	0.23
	SCAD	0.00	0.59	0.00	0.50	0.00	0.20	0.06	0.81	0.81	0.08
	LASSO	0.00	0.64	0.00	0.60	0.00	0.29	0.08	0.81	0.81	0.07
M1/M3	TriSNAR$_G$	0.00	0.07	0.26	0.04	0.23	0.11	0.13	0.85	0.85	0.02
	TriSNAR$_A$	**0.00**	**0.16**	**0.03**	**0.10**	**0.16**	**0.04**	**0.13**	**0.85**	**0.85**	**0.02**
	TLASSO	0.00	0.33	0.00	0.56	0.02	0.40	0.11	0.84	0.85	0.20
	SCAD	**0.00**	**0.23**	**0.00**	**0.15**	**0.15**	**0.04**	**0.13**	**0.84**	**0.85**	**0.07**
	LASSO	0.00	0.33	0.00	0.45	0.04	0.20	0.13	0.85	0.85	0.07
NS1	TriSNAR$_G$	0.00	0.59	0.00	0.24	0.52	0.08	0.09	0.82	0.83	0.02
	TriSNAR$_A$	0.00	0.48	0.01	0.22	0.63	0.09	0.10	0.82	0.83	0.02
	TLASSO	**0.00**	**0.34**	**0.00**	**0.08**	**0.40**	**0.02**	**0.08**	**0.82**	**0.82**	**0.19**
	SCAD	0.00	0.52	0.01	0.18	0.70	0.08	0.10	0.82	0.83	0.10
	LASSO	0.00	0.66	0.00	0.56	0.55	0.37	0.11	0.82	0.83	0.09

Table 12: Simulation $d = 20$ with 3 lags, $T = 100$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.04**	**0.85**	**0.85**	**0.11**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.04**	**0.85**	**0.85**	**0.02**
	TLASSO	0.00	0.00	0.00	0.80	0.00	0.23	0.07	0.85	0.86	0.10
	SCAD	0.00	0.48	0.00	0.62	0.11	0.13	0.06	0.85	0.86	0.17
	LASSO	0.00	0.62	0.00	0.84	0.10	0.29	0.08	0.87	0.88	0.14
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.84**	**0.84**	**0.11**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.84**	**0.84**	**0.02**
	TLASSO	0.01	0.50	0.01	0.91	0.08	0.56	0.08	0.86	0.87	0.10
	SCAD	0.00	0.67	0.00	0.94	0.03	0.55	0.07	0.82	0.85	0.18
	LASSO	0.00	0.67	0.00	0.97	0.01	0.70	0.08	0.84	0.86	0.14
M1	TriSNAR$_G$	0.00	0.00	0.80	0.00	0.66	0.00	0.08	0.84	0.84	0.15
	TriSNAR$_A$	0.00	0.00	0.80	0.00	0.66	0.00	0.08	0.84	0.84	0.03
	TLASSO	**0.00**	**0.00**	**0.06**	**0.44**	**0.35**	**0.15**	**0.09**	**0.84**	**0.85**	**0.12**
	SCAD	0.00	0.57	0.37	0.54	0.64	0.18	0.09	0.83	0.85	0.23
	LASSO	0.00	0.65	0.22	0.69	0.59	0.33	0.11	0.86	0.87	0.20
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.77**	**0.00**	**0.64**	**0.02**	**0.08**	**0.82**	**0.83**	**0.14**
	TriSNAR$_A$	**0.00**	**0.00**	**0.77**	**0.00**	**0.64**	**0.02**	**0.08**	**0.82**	**0.83**	**0.03**
	TLASSO	0.00	0.50	0.28	0.80	0.57	0.50	0.10	0.85	0.85	0.12
	SCAD	**0.00**	**0.66**	**0.06**	**0.78**	**0.48**	**0.45**	**0.10**	**0.80**	**0.84**	**0.20**
	LASSO	0.00	0.67	0.02	0.84	0.41	0.54	0.10	0.82	0.84	0.16
M1/M3	TriSNAR$_G$	**0.14**	**0.00**	**0.82**	**0.01**	**0.80**	**0.01**	**0.11**	**0.87**	**0.88**	**0.10**
	TriSNAR$_A$	**0.14**	**0.00**	**0.82**	**0.01**	**0.80**	**0.01**	**0.11**	**0.87**	**0.88**	**0.02**
	TLASSO	0.50	0.00	0.70	0.24	0.91	0.11	0.11	0.89	0.89	0.12
	SCAD	0.00	0.33	0.19	0.70	0.81	0.55	0.13	0.84	0.90	0.19
	LASSO	0.00	0.33	0.12	0.74	0.73	0.58	0.12	0.87	0.89	0.16
NS1	TriSNAR$_G$	0.00	0.00	0.95	0.00	0.95	0.00	0.10	0.84	0.84	0.16
	TriSNAR$_A$	0.00	0.00	0.95	0.00	0.95	0.00	0.10	0.84	0.84	0.03
	TLASSO	**0.00**	**0.00**	**0.25**	**0.00**	**0.86**	**0.00**	**0.10**	**0.85**	**0.85**	**0.13**
	SCAD	0.00	0.64	0.58	0.42	0.94	0.23	0.11	0.83	0.86	0.21
	LASSO	0.00	0.66	0.43	0.51	0.91	0.33	0.11	0.86	0.87	0.17

Table 13: Simulation $d = 20$ with 3 lags, $T = 200$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	0.00	0.12	0.00	0.14	0.00	0.03	0.04	0.85	0.85	0.04
	TriSNAR$_A$	0.00	0.11	0.00	0.13	0.00	0.03	0.04	0.85	0.85	0.01
	TLASSO	0.00	0.00	0.00	0.31	0.00	0.07	0.06	0.85	0.85	0.16
	SCAD	**0.00**	**0.08**	**0.00**	**0.11**	**0.01**	**0.04**	**0.04**	**0.85**	**0.85**	**0.13**
	LASSO	0.00	0.36	0.00	0.49	0.01	0.20	0.07	0.86	0.87	0.11
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.84**	**0.84**	**0.04**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.84**	**0.84**	**0.02**
	TLASSO	0.00	0.50	0.00	0.94	0.00	0.61	0.06	0.84	0.85	0.15
	SCAD	0.00	0.66	0.00	0.94	0.00	0.52	0.05	0.83	0.84	0.12
	LASSO	0.00	0.67	0.00	0.96	0.00	0.64	0.06	0.84	0.85	0.10
M1	TriSNAR$_G$	**0.00**	**0.02**	**0.10**	**0.01**	**0.17**	**0.11**	**0.07**	**0.83**	**0.83**	**0.06**
	TriSNAR$_A$	0.00	0.06	0.13	0.04	0.21	0.10	0.07	0.83	0.83	0.02
	TLASSO	0.00	0.02	0.00	0.31	0.20	0.09	0.08	0.84	0.84	0.21
	SCAD	0.00	0.25	0.54	0.18	0.63	0.04	0.08	0.84	0.84	0.17
	LASSO	0.00	0.47	0.27	0.37	0.57	0.11	0.10	0.86	0.86	0.15
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.80**	**0.00**	**0.66**	**0.00**	**0.08**	**0.82**	**0.83**	**0.06**
	TriSNAR$_A$	**0.00**	**0.00**	**0.80**	**0.00**	**0.66**	**0.00**	**0.08**	**0.82**	**0.83**	**0.02**
	TLASSO	0.00	0.49	0.10	0.71	0.35	0.34	0.09	0.83	0.84	0.21
	SCAD	0.00	0.65	0.01	0.73	0.29	0.34	0.08	0.80	0.83	0.15
	LASSO	0.00	0.66	0.00	0.79	0.23	0.41	0.08	0.82	0.83	0.13
M1/M3	TriSNAR$_G$	**0.02**	**0.07**	**0.10**	**0.52**	**0.61**	**0.26**	**0.11**	**0.87**	**0.88**	**0.04**
	TriSNAR$_A$	0.02	0.08	0.18	0.43	0.70	0.20	0.12	0.86	0.88	0.02
	TLASSO	0.50	0.00	0.66	0.07	0.86	0.02	0.11	0.89	0.89	0.20
	SCAD	0.00	0.32	0.19	0.52	0.72	0.26	0.12	0.86	0.88	0.15
	LASSO	0.00	0.33	0.09	0.63	0.61	0.35	0.11	0.87	0.88	0.13
NS1	TriSNAR$_G$	0.00	0.03	0.69	0.03	0.87	0.01	0.09	0.83	0.84	0.07
	TriSNAR$_A$	0.00	0.38	0.59	0.17	0.91	0.07	0.10	0.83	0.84	0.02
	TLASSO	**0.00**	**0.03**	**0.02**	**0.00**	**0.80**	**0.00**	**0.09**	**0.83**	**0.83**	**0.21**
	SCAD	0.00	0.55	0.60	0.26	0.93	0.11	0.10	0.83	0.84	0.17
	LASSO	0.00	0.66	0.36	0.37	0.90	0.21	0.11	0.86	0.86	0.15

Table 14: Simulation $d = 20$ with 3 lags, $T = 500$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.85**	**0.85**	**0.03**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.85**	**0.85**	**0.02**
	TLASSO	0.00	0.00	0.00	0.79	0.00	0.21	0.05	0.85	0.85	0.29
	SCAD	0.00	0.11	0.00	0.14	0.00	0.02	0.04	0.85	0.85	0.15
	LASSO	0.00	0.61	0.00	0.81	0.00	0.24	0.06	0.85	0.85	0.14
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**0.03**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**0.02**
	TLASSO	0.00	0.50	0.00	0.92	0.00	0.56	0.05	0.84	0.84	0.30
	SCAD	0.00	0.66	0.00	0.93	0.00	0.53	0.03	0.83	0.84	0.13
	LASSO	0.00	0.66	0.00	0.96	0.00	0.61	0.05	0.84	0.84	0.12
M1	TriSNAR$_G$	**0.00**	**0.13**	**0.06**	**0.14**	**0.05**	**0.17**	**0.06**	**0.82**	**0.83**	**0.06**
	TriSNAR$_A$	**0.00**	**0.20**	**0.02**	**0.17**	**0.05**	**0.15**	**0.06**	**0.82**	**0.83**	**0.03**
	TLASSO	**0.00**	**0.00**	**0.00**	**0.58**	**0.00**	**0.17**	**0.07**	**0.83**	**0.83**	**0.42**
	SCAD	0.00	0.49	0.17	0.45	0.29	0.15	0.07	0.83	0.83	0.22
	LASSO	0.00	0.67	0.00	0.71	0.10	0.28	0.08	0.83	0.84	0.20
M2	TriSNAR$_G$	0.00	0.00	0.08	0.00	0.06	0.13	0.05	0.81	0.81	0.06
	TriSNAR$_A$	**0.00**	**0.01**	**0.07**	**0.00**	**0.06**	**0.13**	**0.05**	**0.81**	**0.81**	**0.03**
	TLASSO	**0.00**	**0.50**	**0.00**	**0.73**	**0.01**	**0.29**	**0.07**	**0.82**	**0.82**	**0.49**
	SCAD	0.00	0.65	0.00	0.75	0.04	0.31	0.06	0.81	0.81	0.20
	LASSO	0.00	0.66	0.00	0.79	0.02	0.35	0.07	0.81	0.82	0.18
M1/M3	TriSNAR$_G$	**0.00**	**0.00**	**0.78**	**0.00**	**0.65**	**0.00**	**0.10**	**0.85**	**0.86**	**0.04**
	TriSNAR$_A$	**0.00**	**0.00**	**0.78**	**0.00**	**0.65**	**0.00**	**0.10**	**0.85**	**0.86**	**0.04**
	TLASSO	0.43	0.00	0.49	0.55	0.55	0.21	0.10	0.87	0.87	0.44
	SCAD	**0.00**	**0.17**	**0.19**	**0.12**	**0.60**	**0.03**	**0.11**	**0.85**	**0.86**	**0.20**
	LASSO	**0.00**	**0.31**	**0.06**	**0.33**	**0.47**	**0.09**	**0.10**	**0.87**	**0.88**	**0.18**
NS1	TriSNAR$_G$	0.00	0.57	0.02	0.17	0.73	0.05	0.08	0.81	0.82	0.06
	TriSNAR$_A$	0.00	0.63	0.03	0.27	0.84	0.12	0.08	0.81	0.82	0.03
	TLASSO	**0.00**	**0.01**	**0.00**	**0.00**	**0.73**	**0.00**	**0.07**	**0.82**	**0.82**	**0.43**
	SCAD	0.00	0.65	0.03	0.42	0.83	0.24	0.09	0.81	0.82	0.23
	LASSO	0.00	0.67	0.01	0.53	0.81	0.35	0.09	0.82	0.83	0.21

Table 15: Simulation $d = 20$ with 3 lags, $T = 1000$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.85**	**0.85**	**0.05**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.85**	**0.85**	**0.04**
	TLASSO	0.00	0.00	0.00	0.16	0.00	0.02	0.05	0.85	0.85	0.47
	SCAD	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.03**	**0.85**	**0.85**	**0.23**
	LASSO	0.00	0.35	0.00	0.38	0.00	0.08	0.06	0.85	0.85	0.22
D2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**0.05**
	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.84**	**0.84**	**0.04**
	TLASSO	0.00	0.49	0.00	0.90	0.00	0.51	0.04	0.84	0.84	0.53
	SCAD	0.00	0.66	0.00	0.93	0.00	0.51	0.03	0.84	0.84	0.19
	LASSO	0.00	0.66	0.00	0.94	0.00	0.57	0.04	0.84	0.84	0.18
M1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.01**	**0.00**	**0.12**	**0.05**	**0.82**	**0.82**	**0.08**
	TriSNAR$_A$	0.00	0.01	0.00	0.02	0.00	0.12	0.05	0.82	0.82	0.05
	TLASSO	0.00	0.12	0.00	0.46	0.00	0.17	0.06	0.83	0.83	0.77
	SCAD	0.00	0.59	0.00	0.41	0.08	0.11	0.06	0.82	0.83	0.37
	LASSO	0.00	0.67	0.00	0.84	0.00	0.51	0.07	0.82	0.83	0.35
M2	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.41**	**0.00**	**0.15**	**0.04**	**0.81**	**0.81**	**0.08**
	TriSNAR$_A$	0.00	0.03	0.00	0.41	0.00	0.15	0.04	0.81	0.81	0.05
	TLASSO	0.00	0.50	0.00	0.69	0.00	0.24	0.06	0.81	0.81	0.99
	SCAD	0.00	0.65	0.00	0.74	0.00	0.30	0.05	0.81	0.81	0.33
	LASSO	0.00	0.65	0.00	0.76	0.00	0.31	0.06	0.81	0.81	0.31
M1/M3	TriSNAR$_G$	**0.00**	**0.12**	**0.01**	**0.08**	**0.02**	**0.13**	**0.09**	**0.84**	**0.85**	**0.05**
	TriSNAR$_A$	**0.00**	**0.24**	**0.00**	**0.31**	**0.12**	**0.07**	**0.09**	**0.84**	**0.85**	**0.04**
	TLASSO	0.01	0.29	0.08	0.68	0.22	0.48	0.09	0.85	0.85	0.90
	SCAD	0.00	0.32	0.00	0.51	0.13	0.13	0.10	0.84	0.85	0.30
	LASSO	0.00	0.33	0.00	0.72	0.05	0.28	0.09	0.85	0.86	0.28
NS1	TriSNAR$_G$	**0.00**	**0.00**	**0.00**	**0.00**	**0.74**	**0.00**	**0.07**	**0.82**	**0.82**	**0.09**
	TriSNAR$_A$	0.00	0.46	0.01	0.11	0.83	0.04	0.07	0.82	0.82	0.05
	TLASSO	0.00	0.26	0.00	0.04	0.67	0.01	0.06	0.81	0.82	0.77
	SCAD	0.00	0.66	0.01	0.36	0.83	0.19	0.08	0.81	0.82	0.38
	LASSO	0.00	0.67	0.00	0.55	0.77	0.37	0.08	0.82	0.82	0.36

Table 16: Simulation $d = 100$ with 3 lags, $T = 100$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_A$	**0.00**	**0.63**	**0.00**	**0.88**	**0.08**	**0.19**	**0.03**	**0.86**	**0.86**	**7.08**
	TLASSO	0.00	0.67	0.00	1.00	0.00	1.00	0.27	0.36	1.21	3.97
	SCAD	0.00	0.67	0.00	0.97	0.22	0.37	0.03	0.87	0.87	1.31
	LASSO	0.00	0.67	0.00	0.98	0.19	0.50	0.04	0.88	0.88	1.37
D2	TriSNAR$_A$	**0.00**	**0.67**	**0.00**	**0.99**	**0.05**	**0.63**	**0.04**	**0.86**	**0.87**	**1.77**
	TLASSO	0.00	0.67	0.00	1.00	0.00	1.00	0.27	0.35	1.21	4.04
	SCAD	0.00	0.67	0.00	0.99	0.07	0.74	0.05	0.87	0.89	1.31
	LASSO	0.00	0.67	0.00	1.00	0.05	0.80	0.04	0.86	0.87	1.38
M1	TriSNAR$_A$	0.00	0.66	0.22	0.89	0.59	0.36	0.05	0.83	0.86	11.05
	TLASSO	**0.00**	**0.51**	**0.01**	**0.89**	**0.14**	**0.77**	**0.22**	**0.53**	**1.07**	**4.15**
	SCAD	0.00	0.67	0.08	0.94	0.66	0.52	0.05	0.82	0.86	1.50
	LASSO	0.00	0.67	0.05	0.94	0.65	0.52	0.05	0.86	0.88	1.50
M2	TriSNAR$_A$	**0.00**	**0.65**	**0.05**	**0.91**	**0.53**	**0.53**	**0.05**	**0.79**	**0.85**	**2.33**
	TLASSO	0.00	0.66	0.01	0.98	0.02	0.98	0.25	0.38	1.13	4.19
	SCAD	0.00	0.67	0.00	0.97	0.53	0.76	0.07	0.76	0.87	1.52
	LASSO	0.00	0.67	0.00	0.97	0.53	0.72	0.05	0.82	0.85	1.53
M1/M3	TriSNAR$_A$	**0.00**	**0.33**	**0.08**	**0.92**	**0.85**	**0.76**	**0.07**	**0.82**	**0.91**	**1.50**
	TLASSO	0.03	0.32	0.03	0.96	0.04	0.96	0.25	0.46	1.15	3.94
	SCAD	0.00	0.33	0.06	0.95	0.87	0.83	0.07	0.81	0.92	1.39
	LASSO	0.00	0.33	0.04	0.95	0.86	0.83	0.06	0.86	0.89	1.42
NS1	TriSNAR$_A$	**0.00**	**0.66**	**0.70**	**0.34**	**0.99**	**0.19**	**0.05**	**0.82**	**0.86**	**10.67**
	TLASSO	0.00	0.67	0.00	0.67	0.00	0.67	0.25	0.39	1.12	4.05
	SCAD	0.00	0.67	0.53	0.54	0.99	0.38	0.06	0.81	0.86	1.51
	LASSO	0.00	0.67	0.49	0.55	0.98	0.38	0.05	0.86	0.88	1.51

Table 17: Simulation $d = 100$ with 3 lags, $T = 200$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**2.30**
	TLASSO	0.00	0.00	0.00	0.97	0.00	0.37	0.03	0.85	0.85	7.79
	SCAD	0.00	0.26	0.00	0.33	0.03	0.01	0.02	0.85	0.85	4.03
	LASSO	0.00	0.55	0.00	0.74	0.03	0.05	0.04	0.87	0.87	3.62
D2	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.01**	**0.84**	**0.84**	**3.35**
	TLASSO	0.00	0.50	0.00	0.97	0.00	0.30	0.03	0.86	0.87	8.00
	SCAD	0.00	0.67	0.00	0.99	0.00	0.65	0.04	0.82	0.86	4.11
	LASSO	0.00	0.67	0.00	0.99	0.00	0.75	0.03	0.84	0.85	3.71
M1	TriSNAR$_A$	0.00	0.00	0.80	0.00	0.67	0.00	0.03	0.84	0.84	3.52
	TLASSO	**0.00**	**0.00**	**0.09**	**0.04**	**0.61**	**0.00**	**0.05**	**0.86**	**0.86**	**8.78**
	SCAD	0.00	0.63	0.14	0.74	0.62	0.18	0.04	0.83	0.84	4.22
	LASSO	0.00	0.64	0.09	0.73	0.61	0.14	0.05	0.87	0.87	3.86
M2	TriSNAR$_A$	**0.00**	**0.00**	**0.77**	**0.00**	**0.64**	**0.02**	**0.03**	**0.82**	**0.83**	**5.10**
	TLASSO	0.00	0.50	0.53	0.91	0.66	0.24	0.05	0.86	0.86	9.08
	SCAD	0.00	0.67	0.00	0.97	0.33	0.72	0.06	0.78	0.84	4.25
	LASSO	0.00	0.67	0.00	0.97	0.35	0.59	0.04	0.82	0.83	3.89
M1/M3	TriSNAR$_A$	**0.07**	**0.00**	**0.81**	**0.01**	**0.77**	**0.00**	**0.05**	**0.87**	**0.88**	**2.58**
	TLASSO	0.50	0.00	0.57	0.68	0.83	0.21	0.05	0.89	0.89	8.13
	SCAD	0.00	0.33	0.04	0.92	0.76	0.59	0.06	0.84	0.89	4.09
	LASSO	0.00	0.33	0.02	0.94	0.73	0.63	0.05	0.87	0.89	3.69
NS1	TriSNAR$_A$	0.00	0.00	0.99	0.00	0.99	0.00	0.04	0.84	0.84	3.71
	TLASSO	**0.00**	**0.00**	**0.01**	**0.00**	**0.96**	**0.00**	**0.04**	**0.82**	**0.83**	**8.39**
	SCAD	0.00	0.67	0.51	0.36	0.98	0.21	0.05	0.84	0.84	4.31
	LASSO	0.00	0.67	0.44	0.36	0.98	0.20	0.05	0.87	0.87	3.90

Table 18: Simulation $d = 100$ with 3 lags, $T = 500$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**1.56**
	TLASSO	0.00	0.00	0.00	0.34	0.00	0.01	0.03	0.85	0.85	11.86
	SCAD	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.85	0.85	6.95
	LASSO	0.00	0.67	0.00	0.98	0.00	0.45	0.03	0.85	0.85	6.03
D2	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.01**	**0.84**	**0.84**	**1.75**
	TLASSO	0.00	0.50	0.00	0.90	0.00	0.13	0.03	0.85	0.85	12.79
	SCAD	0.00	0.67	0.00	0.99	0.00	0.54	0.02	0.83	0.84	7.26
	LASSO	0.00	0.67	0.00	0.99	0.00	0.68	0.02	0.84	0.84	6.32
M1	TriSNAR$_A$	0.00	0.07	0.19	0.03	0.21	0.10	0.03	0.83	0.83	3.76
	TLASSO	**0.00**	**0.00**	**0.00**	**0.71**	**0.02**	**0.05**	**0.04**	**0.83**	**0.83**	**16.19**
	SCAD	0.00	0.50	0.18	0.34	0.62	0.05	0.03	0.84	0.84	8.19
	LASSO	0.00	0.50	0.05	0.52	0.39	0.18	0.04	0.85	0.85	7.33
M2	TriSNAR$_A$	**0.00**	**0.00**	**0.80**	**0.00**	**0.67**	**0.00**	**0.03**	**0.82**	**0.82**	**3.91**
	TLASSO	0.00	0.50	0.01	0.58	0.46	0.06	0.04	0.84	0.84	18.27
	SCAD	0.00	0.67	0.00	0.98	0.04	0.73	0.05	0.80	0.82	8.49
	LASSO	0.00	0.67	0.00	0.96	0.05	0.45	0.03	0.82	0.82	7.56
M1/M3	TriSNAR$_A$	**0.00**	**0.33**	**0.01**	**0.70**	**0.58**	**0.12**	**0.05**	**0.85**	**0.86**	**3.40**
	TLASSO	0.50	0.00	0.55	0.38	0.76	0.03	0.05	0.88	0.88	15.03
	SCAD	0.00	0.33	0.02	0.76	0.59	0.17	0.05	0.85	0.86	8.20
	LASSO	0.00	0.33	0.00	0.83	0.54	0.20	0.05	0.88	0.88	7.12
NS1	TriSNAR$_A$	0.00	0.66	0.01	0.33	0.97	0.17	0.04	0.81	0.81	3.01
	TLASSO	**0.00**	**0.03**	**0.00**	**0.00**	**0.95**	**0.00**	**0.04**	**0.82**	**0.82**	**14.86**
	SCAD	0.00	0.65	0.23	0.32	0.97	0.21	0.04	0.81	0.82	8.13
	LASSO	0.00	0.67	0.00	0.54	0.96	0.35	0.04	0.82	0.83	7.10

Table 19: Simulation $d = 100$ with 3 lags, $T = 1000$ for D1, D2, M1, M2, M1/M3, NS1. The best performing model in terms of lowest FN and FP for each number of observations is indicated in bold.

		FN.l	FP.l	FN.g	FP.g	FN.e	FP.e	MAE.para	MAE.res	MAFE.res	time (in s)/combination
D1	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.85**	**0.85**	**2.11**
	TLASSO	0.00	0.01	0.00	0.98	0.00	0.38	0.02	0.85	0.85	30.18
	SCAD	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.85	0.85	8.62
	LASSO	0.00	0.55	0.00	0.66	0.00	0.06	0.03	0.85	0.85	7.66
D2	TriSNAR$_A$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.01**	**0.84**	**0.84**	**2.56**
	TLASSO	0.00	0.49	0.00	0.85	0.00	0.09	0.02	0.84	0.84	32.85
	SCAD	0.00	0.67	0.00	0.99	0.00	0.54	0.01	0.84	0.84	8.39
	LASSO	0.00	0.67	0.00	0.99	0.00	0.64	0.02	0.84	0.84	7.44
M1	TriSNAR$_A$	**0.00**	**0.59**	**0.00**	**0.41**	**0.00**	**0.14**	**0.02**	**0.82**	**0.82**	**4.50**
	TLASSO	0.00	0.02	0.00	0.82	0.00	0.37	0.03	0.83	0.83	47.46
	SCAD	**0.00**	**0.46**	**0.24**	**0.25**	**0.64**	**0.03**	**0.03**	**0.84**	**0.84**	**14.58**
	LASSO	0.00	0.67	0.00	0.81	0.03	0.15	0.04	0.83	0.84	13.52
M2	TriSNAR$_A$	**0.00**	**0.01**	**0.00**	**0.01**	**0.00**	**0.06**	**0.02**	**0.81**	**0.81**	**4.17**
	TLASSO	0.00	0.49	0.00	0.69	0.01	0.25	0.03	0.82	0.82	57.68
	SCAD	0.00	0.67	0.00	0.98	0.00	0.76	0.04	0.80	0.82	11.25
	LASSO	0.00	0.67	0.00	0.95	0.00	0.39	0.03	0.81	0.81	10.39
M1/M3	TriSNAR$_A$	**0.00**	**0.00**	**0.71**	**0.00**	**0.62**	**0.00**	**0.04**	**0.85**	**0.85**	**2.57**
	TLASSO	0.50	0.00	0.50	0.09	0.66	0.00	0.05	0.88	0.88	48.72
	SCAD	0.00	0.32	0.01	0.34	0.51	0.03	0.05	0.85	0.85	10.41
	LASSO	0.00	0.32	0.00	0.38	0.46	0.02	0.05	0.87	0.88	9.43
NS1	TriSNAR$_A$	0.00	0.60	0.00	0.10	0.97	0.04	0.04	0.81	0.81	2.97
	TLASSO	**0.00**	**0.00**	**0.00**	**0.00**	**0.94**	**0.00**	**0.03**	**0.81**	**0.81**	**41.72**
	SCAD	0.00	0.66	0.00	0.51	0.96	0.29	0.04	0.81	0.81	11.18
	LASSO	0.00	0.67	0.00	0.49	0.96	0.30	0.04	0.82	0.82	10.17