# Joint Estimation of Multiple Granger Graphical Models using Non-convex Penalties

## Thesis Proposal

Parinthorn Manomaisaowapak

Advisor: Assistant Professor Dr. Jitkomut Songsiri

Department of Electrical Engineering
Faculty of Engineering

Chulalongkorn University

May 26, 2020

# Contents

### Abstract

Learning causality among variables in multivariate time-series is a way to characterize the cause and effect relationship as a causality network. In multiple multivariate time-series, we assume that the network of each multivariate time-series decomposes into two parts, the global part, and the local part. The global part is a part where all networks share the common connections, while the local part is the leftover connections in each network. Granger causality quantifies the causality network using a vector autoregressive (VAR) model. The joint estimation of multiple sparse Granger causality networks is estimated by using a regularization term as a convex penalty such as group lasso penalty. Recent advancements in the sparse estimation field suggested that the non-convex penalties can be used to replace the convex penalties because the non-convex penalties have a better sparsity recovery rate than the convex penalties. This proposal considers an extension of joint estimation formulations of multiple sparse Granger causality networks using non-convex group norm penalties. The non-convex group norm penalties have a well-established statistical property that is shown to be superior to a group lasso penalty. Since the problem is non-convex non-smooth, we applied the state-of-the-art, non-monotone accelerated proximal gradient (nmAPG), algorithm to solve one of our formulations in a large scale setting and we fine-tuned penalty parameter in the ADMM (Alternating Direction Methods of Multipliers) algorithm to solve other formulations. The joint estimation with the non-convex group norm penalty outperformed joint estimation based on group lasso penalty in our intensive simulated data experiments.

## 1 Introduction

Granger causality (GC) is used to quantify the strength of a cause and effect between time-series. The Granger causality between all pairs of time-series forms a network call Granger causality network or a GC network. The causality connections between multiple time-series provide us a deeper understanding of their nature and their relationships. For instance, human brain functionality can be described by a measure called effective brain connectivity. The effective brain connectivity quantified the amount of information that one brain region exerts to another [Fri11]. These relations belong in the cause-and-effect type of relationships, which can be investigated by Granger causality. A Granger causality network among multiple time-series can be determined by a linear vector autoregressive model (VAR). This model can be efficiently estimated by an ordinary linear least-squares method or by solving the Yule-Walker equation [Lüt05]. These methods generally produce a dense GC network. A lower density GC network or a sparse GC network is preferred because it signifies the importance of those few connections. A sparse GC network can be obtained by solving a regularized least-square using a group lasso penalty [Hau12], [Son13].

The identification of a sparse GC network from a single multivariate time-series is only a local method to determine the GC network. It is not natural to use a local method to infer on the global GC network. For example, in neuroscience applications, the human brain network cannot be concluded from a single subject, but it is possible to have a better approximation of the global brain network from multiple subjects, which is called group-level brain connectivity. However, the global network can be approximated from multiple networks. The joint estimation of multiple Granger networks is the method that can be used to obtain an estimated version of the global network using multiple networks. This method controls multiple GC networks to have a similar sparsity structure. The similarity in all GC networks forms an approximation of the global GC network. With the presence of heterogeneity in each network, the network can be decomposed into two parts, the global network and the local network. We will refer to the global part as a common GC network or a homogeneous part of the GC network. The local part is the connections of each network that is different from the common GC network. We will refer this part as a differential GC network or a heterogeneous part of the GC network. The general form of jointly sparse causality networks estimation is usually defined by the optimization problem,

$$\underset{x}{\text{minimize}} \;\; g(x_1, x_2, ..., x_K) + \sum_{i=1}^{K} [f(x_i) + h(x_i)] \tag{1}$$

where $f$ is a smooth loss function, which depends on the framework. The term $h$ is a sparsity inducing the regularization of a local network, and the sparsity of the global network is regularized by term $g$. Both $g$ and $h$ have gradient discontinuity at $x = 0$. The problem parameters are $x_1, \dots, x_K$ where each $x_i$ is the parameter of $i$th time-series model. The formulation (1) has already been applied in a Gaussian graphical model (GGM) framework where $x_i$ is an inverse covariance matrix [TB20]. This formulation has already been applied to jointly estimate multiple sparse GC networks [Son17, SM19a, SM19b, GKMM15, WBC18]. For example, [Son17] used a group fused lasso penalty as the term $g$ and group lasso for the term $h$. [SM19a] proposed a fused lasso penalty as $g$ and a lasso penalty as $h$ to jointly estimate the model. In [SM19b], they proposed to use group lasso to extract the homogeneous part of the GC matrix from multiple data and use this model to estimate the heterogeneous part, which is a multi-stage optimization approach.

These methods rely on non-smooth convex penalties or lasso-type penalties. It is known that these convex penalties have an estimation bias problem [WCLQ18]. The lasso-type penalty is known that it shrinks variables equally in all magnitude, so the estimated variables are biased toward zero. In general, the bias of convex penalties is fixed by using a constraint least-square to re-fit the models with the sparsity pattern learned from regularized regression using the convex penalty. This approach is not possible when the system of linear equation is an under-determined system that occurred when the problem is in a high-dimensional setting. In other cases, the non-convex penalties such as $\ell_q$ penalty, with $0 < q < 1$, SCAD penalty are known to have a better recovery rate than the $\ell_1$ penalty. So, the replacement of the group lasso by a non-convex group penalty is a natural extension to the jointly sparse GC estimation. The non-convex $\ell_{p,q}$ group norm penalty

$$\|x\|_{p,q} = \sum_{i=1}^{K} \|x_i\|_p^q$$

with $p \geq 1$ and $0 < q < 1$, has been proposed in [HLM$^+$17] with theoretical results that it has a better sparsity recovery rate than the group lasso penalty. In the GGM framework, the non-convex regularization has already been applied. The group norm penalty that is used to regulate the sparsity of the variables across multiple groups was originally introduced as a group bridge penalty [HMXZ09]. In [GLMZ11], they proposed an $\ell_{p,q}$ penalty with, $p = 1, q = 1/2$. They claimed that this penalty has a representation as hierarchical regularization that decomposed sparsity structure into a global part and local part. In [CZZ15], they also provided a generalization of [GLMZ11] and two more non-convex functions that also have representation as hierarchical regularization. Recently, the estimation of a single Granger graphical model with a non-convex penalty and $\ell_1$ type loss function has been proposed in [BLH$^+$20]; however, they do not use a group norm penalty to exploit the relation between VAR model's coefficients and Granger causality.

The non-convex extension turns the simple-to-solve convex problem to a non-convex optimization problem. The gradient discontinuity at zero of the terms $g, h$ in (1) forbids the usage of gradient-based methods since the optimal point is expected to be precisely zero. An efficient algorithm is required to solve the problem in a large scale setting. Back to the application, the inference task requires the sparse estimation method to eliminate insignificant causality. A thresholding operator should be involved in the optimization process. The threshold is used to distinguish between a true zero estimated variable and a small non-zero valued variable. In a large-scale setting, a first-order algorithm is preferred because second-order methods are not feasible to use in a high-dimensional case. The available first-order methods that have a thresholding operation in the convex cases are the proximal algorithms [PB14], such as proximal gradient methods and ADMM (Alternating Direction Methods of Multipliers) algorithm [BPC$^+$11]. The proximal algorithms are one of the non-smooth optimization methods for solving the original problem directly without smoothing the non-smooth part. When $g, h$ in (1) are non-convex regularization functions, the available algorithms are limited because the formulation property violates the sufficient condition of many existing algorithms, such as the variants of the ADMM algorithm. The convergence analysis of a general descent method for problems that similar to the problem (1) is presented in [ABS13]. Their analysis suggested that the algorithm should generate a strictly decreasing sequence to sufficiently obtain a global convergence. The proximal gradient algorithm is one of the descent algorithms which can be employed to solve the problem (1). However, the proximal gradient algorithm requires a proximal operator of the regularization terms in (1). When the closed-form expression of the proximal operator is available, the state-of-the-art algorithm, non-monotone proximal gradient algorithm [LL15], can be employed to solve (1). In general, the term $g(x) + \sum_i h(x_i)$ usually does not have a closed-form expression, but each $g, h$ has closed-form proximal operator so the splitting algorithm, such as the ADMM algorithm, can be employed. The challenge in the non-convex case is that the problem (1) directly violates an assumption in the convergence analysis for the ADMM algorithm. Even though the algorithm is failing to converge globally, the convergence of ADMM in our case can be controlled through the ADMM penalty parameter, which can be fine-tuned to obtain convergence to a critical point.

In this proposal, we propose to use a non-convex group norm penalty proposed in [HLM$^+$17] as a non-convex extension to the jointly sparse estimation of Granger networks from multiple multivariate time-series data. Our work extended the original work of [Son17] to the non-convex joint estimation of the Granger causality framework as our main contribution. We also provided an efficient numerical method for solving (1) in a large scale setting using state-of-the-art nmAPG algorithm when $h = 0$. In the case when the terms $g, h$ present in the models, the convergence of the ADMM algorithm can be obtained by fine-tuning the ADMM penalty parameter.

We provide an overview of the work plan in section 2. In section 3, we introduced the concept of Granger causality and its relation to the VAR models. In section 4 is our methodology, which contains the formulations we proposed based on (1). The algorithms are discussed in section 5. The preliminary results and our future works are given at the end of this proposal, which are section 6 and 7, respectively.

# 2 Proposal overview

## 2.1 Objectives

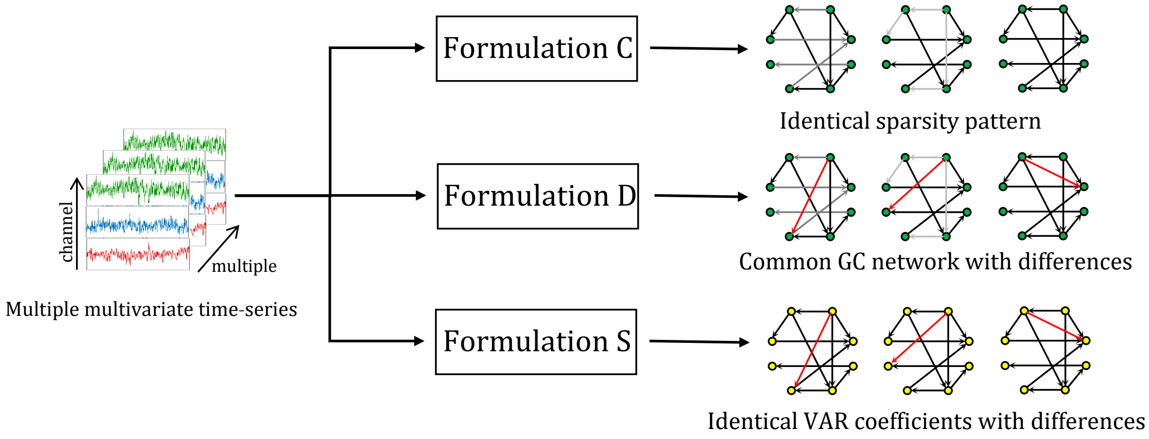

channel
multiple
Multiple multivariate time-series
Formulation C
Formulation D
Formulation S
Identical sparsity pattern
Common GC network with differences
Identical VAR coefficients with differences

Figure 1: Our formulations on different type of multiple Granger networks

1. We aim to provide three formulations presented in Figure 1 which can be used to jointly estimate multiple Granger causality networks based on different assumptions. The formulations are

   (a) Formulation C: The estimated networks have an identical sparsity pattern.

   (b) Formulation D: The estimated networks have some identical parts and some different parts.

   (c) Formulation S: The estimated networks have some block-identical value of VAR's coefficients and some different sparsity pattern.

2. We provide efficient numerical methods to solve the proposed estimation methods in a large scale setting.

## 2.2 Scope of work

1. The proposed framework will be verified on intensive simulations and one real-world data set.

2. The usefulness of the methods will be illustrated on brain network application.

## 2.3 Expected outcome

1. Estimation formulations of multiple Granger graphical models.

2. Computer program that has input as a set of multivariate time-series and return group and individual Granger graphical model of the multiple time-series.

## 2.4 Work plan

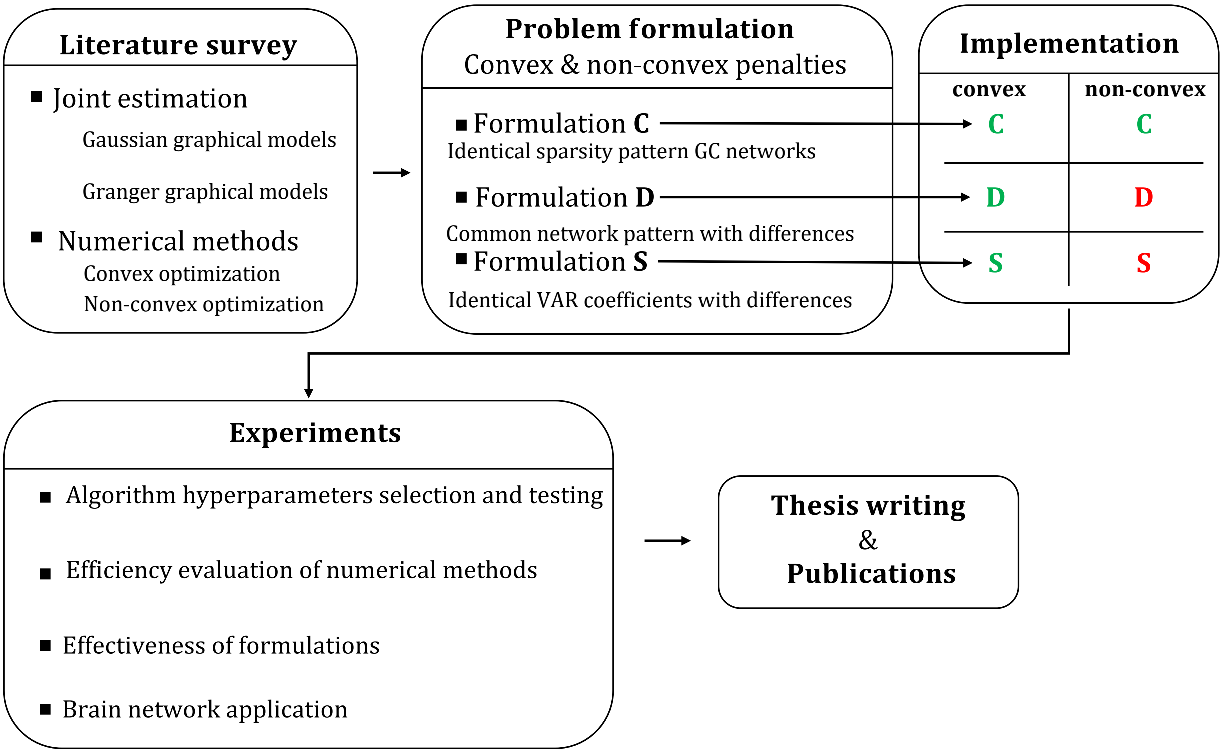

Literature survey
Joint estimation
Gaussian graphical models
Granger graphical models
Numerical methods
Convex optimization
Non-convex optimization
Problem formulation
Convex & non-convex penalties
Formulation C
Identical sparsity pattern GC networks
Formulation D
Common network pattern with differences
Formulation S
Identical VAR coefficients with differences
Implementation
convex
non-convex
C
C
D
D
S
S
Experiments
Algorithm hyperparameters selection and testing
Efficiency evaluation of numerical methods
Effectiveness of formulations
Brain network application
Thesis writing
&
Publications

Figure 2: Thesis work plan diagram. The formulations highlighted in green in the implementation block indicate that the efficient algorithm is available. Formulations D and S in non-convex case do not have an efficient algorithm yet.

# 3 Granger causality estimation

We consider the vector autoregressive (VAR) model

$$y(t) = A_1 y(t-1) + A_2 y(t-2) + \cdots + A_p y(t-p) \tag{2}$$

where $y(t) = (y_1(t), y_2(t), \ldots, y_n(t)) \in \mathbf{R}^n, A_r \in \mathbf{R}^{n \times n}$ is VAR coefficient matrix of lag $r$, $p$ is the model order.The matrix $(A_r)_{ij}$ directly indicates a linear gain from time-series $y_j$ to $y_i$. The Granger causality can be quantified by this gain.

## 3.1 Granger causality

Granger causality (GC) is widely used in linear time-series modeling. A Granger causality between a pair of variables in multivariate time-series can be quantified by estimating one of which with and without another multivariate time-series. If the estimation is better in the sense of having a smaller residual covariance matrix, then there exists a GC connection from the additional time-series to the estimated time-series. The strength of GC connection from $j$th multivariate time-series to $i$th multivariate time-series is quantified by

$$\mathcal{F}_{ij} = \log\det \frac{\Sigma_{ii}^R}{\Sigma_{ii}} \tag{3}$$

where $\Sigma_{ii}$ is the covariance matrix of residuals from the estimation of $i$th multivariate time-series and $\Sigma_{ii}^R$ is the covariance matrix from the estimation of $i$th multivariate time-series without $j$th multivariate time-series. In a study of causality from one-dimensional time-series to another one-dimensional time-series, the residual covariance matrix, $\Sigma_{ii}$, reduced to a variance of $i$th time-series residual.

In VAR model, the necessary and sufficient condition for the absence of causal relation is

$$\mathcal{F}_{ij} = 0 \leftrightarrow (A_r)_{ij} = 0, r = 1, 2, \cdots, p. \tag{4}$$

This condition can be used as an estimation prior knowledge for knowing which VAR coefficients should not exist if there is no GC connection. Both (3) and (4) are characterized in the time domain. In the spectral domain, the Granger causality is defined in the same sense. The spectral decomposition is used to compare the average power of log ratio of reduced model and full model [BS14] as

$$\mathcal{F}_{ij} = \frac{1}{2\pi}\int_0^{2\pi} \log \frac{|S_{ii}(\lambda)|}{|S_{ii}(\lambda) - H_{ij}(\lambda) S_{j|i}(\lambda) H_{ij}^*(\lambda)|} \mathrm{d}\lambda \tag{5}$$

where $S(\lambda) = H(\lambda)\Sigma H(\lambda)^*$ is the spectral decomposition of the VAR process, $S_{ij}(\lambda)$ denotes a $(i,j)$ sub-block of $S(\lambda)$. The spectral definition of GC is also equivalence to transfer entropy when the evaluating variables are Gaussian random variables [BBS09]. These interpretations of Granger causality can be applied to other types of models including the models that are more general than the VAR model such as the state-space model[BS15]. One way to describe the causal relations among multiple time-series is to form it as a network or as a matrix as shown in Figure 3. The yellow entries embedded Granger causality from node $j$ to node $i$. We will refer this as a Granger network, or a GC matrix.

In this kind of causality, the statistical test is required to test whether the causality is significant. A traditional method is to perform the statistical test on each $\mathcal{F}_{ij}$ [BS14] independently. This method can be used either when the sample size is large enough to have an approximated

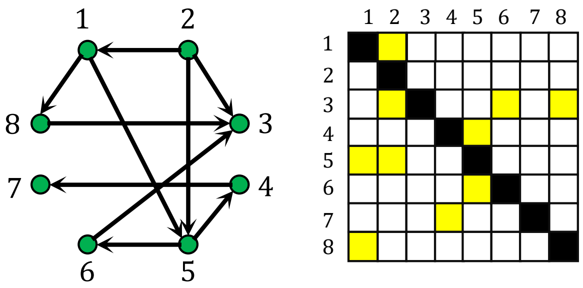


Figure 3: Granger causality network as a graph(left) and as a matrix (right).

asymptotic distribution or when non-parametric methods are applicable. However, each index in the GC matrix should not be treated as if they are independent of each other because they may be related to each other. So, we consider a sparsity prior for estimating the GC matrix instead of performing independent statistical testing. The causality from $j$th time-series to $i$th time-series in VAR models is characterized by (4). This condition motivates us to use a sparse estimation scheme that automatically induces block sparsity in VAR coefficients to have a sparse GC matrix. In general, the sparse estimation of a model can be expressed as

$$\underset{x}{\text{minimize}} \;\; f(x) + g(x)$$

where function $f$ indicates the fitting between model output and the data. The function $g$ is the sparse inducing regularization term. For our model, the fitting and regularization term will be more precisely stated in the following section.

## 3.2 VAR estimation

In VAR estimation, we consider the Ordinary least square (OLS) estimation. The $T$ time points VAR process can be rearranged as

$$\begin{bmatrix} y(p+1) & y(p+2) & \dots & y(T) \end{bmatrix} = \begin{bmatrix} A_1 & \dots & A_p \end{bmatrix} \begin{bmatrix} y(p) & y(p+1) & \dots & y(T-1) \\ \vdots & \vdots & \dots & \vdots \\ y(2) & y(3) & \dots & y(T-p+1) \\ y(1) & y(2) & \dots & y(T-p) \end{bmatrix}$$

which is in the form of $Y = AH$, $Y \in \mathbf{R}^{n\times(T-p)}$, $H \in \mathbf{R}^{np\times(T-p)}$, $A_r \in \mathbf{R}^{n\times n}$ where $n$ is the dimension of time series and $T$ is number of data points, $p$ is the model order. The least-squares estimation of VAR model is

$$\underset{A}{\text{minimize}} \;\; (1/2)\|Y - AH\|_F^2$$

The analytical solution $\hat{A}$ can be obtained by solving the normal equation,

$$\hat{A}(HH^T) = AH^T$$

The least-square solution $\hat{A}$ generally does not produce a sparse solution. The sparse regularization term can be added to the objective to induce sparsity. In our framework, the sparsity pattern of VAR coefficients must follow (4) in order to obtain a sparse GC matrix.

### 3.3 Group norm penalty

From (4), the sparsity in the GC matrix can be enforced through the sparsity in all lags of VAR coefficients. A group norm penalty is capable of shrinking all lags of VAR coefficients to be zero simultaneously in a regularized regression problem. If we denote $x = (x_1, x_2, ..., x_K)$ and each $x_i$ is a sub-block of $x$. The group norm penalty is defined as

$$\|x\|_{p,q} = \sum_{i=1}^{K} \|x_i\|_p^q$$

The group norm penalty is a composition of two different types of norms. For example, a group lasso penalty is a special case of group norm penalty with $p = 2, q = 1$. The summation of a norm is equivalent to find an $\ell_1$ norm of a vector $v = (\|x_1\|_2, \dots, \|x_K\|_2)$. Since $\ell_1$ penalty is known to produce a sparse result for a regularized linear regression model, the problem,

$$\underset{x}{\text{minimize}} \;\; \|Ax - b\|_2^2 + \lambda \sum_{i=1}^{K} \|x_i\|_2, \tag{6}$$

generally produces a sparse result in $v = (\|x_1\|_2, ..., \|x_K\|_2)$. The sparsity in each $v_i$ infers that $x_i$ is a zero vector from the definition of a norm. In other words, the vector $x$ is expected to be a block-sparse vector.

If the block size of variables is equal to $B$, a regularization term is compactly defined as

$$g(x) = \lambda \|Lx\|_{2,1}^{(B)} \tag{7}$$

where $L$ is a designed matrix that choose which linear function of $x$ to be sparse. We provide two examples of explaining the group norm penalty (7) in the following.

- If $x \in \mathbf{R}^{100}, B = 5, L = I$ then $x = (x_1, ..., x_{20}), x_i \in \mathbf{R}^5$ and $\|x\|_{p,q}^{(B)} = \sum_{i=1}^{20} \|x_i\|_p^q$.
- If $x \in \mathbf{R}^{100}, B = 25, L = I$ then $x = (x_1, ..., x_4), x_i \in \mathbf{R}^{25}$ and $\|x\|_{p,q}^{(25)} = \sum_{i=1}^{4} \|x_i\|_p^q$.

If the structural sparsity of model is known, one can design a matrix $L$ to force $Lx$ to be sparse. For instance, if $L$ is a projection matrix, some part of $x$ can be prevented to be zero. If $L$ is a difference matrix, the some adjacent sub-vector of estimated $x$ will be identical to each other. This instance, when the difference matrix is used, is called a group fused lasso [ABD13] penalty. For example when $x = (x_1, x_2, x_3)$, these two cases are

$$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & I \end{bmatrix} x := L_1 x \tag{8}$$

and

$$\begin{bmatrix} x_1 - x_2 \\ x_1 - x_3 \\ x_2 - x_3 \end{bmatrix} = \begin{bmatrix} I & -I & 0 \\ I & 0 & -I \\ 0 & I & -I \end{bmatrix} x := L_2 x \tag{9}$$

The matrix $L_2$ is a difference matrix that takes every combination of differences into account.

The reason behind the sparsity inducing of the property of $\ell_1$ type penalty is because of the feasible region of the equivalent epigraph form has sharp edges along axes. For instance, the lasso regression problem is to solve

$$\text{minimize} \; \|Ax - b\|_2^2 + \lambda \|x\|_1$$

which its equivalent epigraph form of lasso problem is

$$\begin{array}{ll}\text{minimize} & \|Ax-b\|_2^2 \\ \text{subject to} & \|x\|_1 \leq t\end{array}$$

where $\lambda$ is a tuning parameter related to $t$. The feasible region of the lasso problem is an $\ell_1$ norm ball which is a diamond shape with sharp edges along axes. This is for pointing out that not all pairs of $p, q$ of group norm regularization problem are able to produce a sparse result. For example, when $p = q = 2$, the problem becomes a Tikhonov regularized least-squares, which does not produce a sparse result in general. The epigraph form of this problem is

$$\begin{array}{ll}\text{minimize} & \|Ax-b\|_2^2 \\ \text{subject to} & \|x\|_2 \leq t\end{array}$$

This is a direct consequent from its equivalent epigraph form that its feasible region is a sphere which does not have a sharp edge along axes.

In the non-convex penalty $\ell_q$ case, the feasible region in its epigraph form has sharper edges along axes than the lasso-type penalty. Consequently, it is more likely that this penalty will produce sparser results. Moreover, it provides superior theoretical property. In [HLM$^+$17], they proposed a group norm penalty

$$g(x) = \lambda\|x\|_{p,q}^{(B)} = \lambda\sum_{i=1}^{K}\|x_i\|_p^q \tag{10}$$

They also proved that the non-convex group norm penalty is easier to satisfy the restricted eigenvalue condition (REC), which is a condition that indicates the accuracy of a sparse estimation method, than the group lasso penalty. By following their notation, we will refer to this group norm penalty as $\ell_{p,q}$ penalty.

# 4 Methodology

Sparse GC networks can be efficiently estimated with a joint estimation framework, as described in diagram 4. The process begins by feeding $K$ sets of multiple multivariate to the joint estimation process. The joint estimation process embeds the prior information to the optimization problem with tuning parameters as the strength of prior information. The selection of these tuning parameters is discussed at the end of this section.

In this work, we consider three types of prior information. The first prior information that we consider is the case when all $K$ Granger networks have identical sparsity patterns denoted as **common GC network**, as shown in the left path of diagram 4. We refer to this formulation as **formulation C**. The extraction of a group-level effective brain network from multiple sets of fMRI time-series can be one of the applications of this formulation. In general, $K$ ground-truth networks have two parts, the homogeneous part, and the heterogeneous part. Formulation C is able to capture only the homogeneous part but not the heterogeneous part. We then propose the second formulation, **formulation D**, to promote **common & differential network**, which is shown in the central path of the diagram. This formulation adds the degree of freedom to the models by allowing some differences in the networks. The last formulation is called **formulation S**. This formulation decomposed the learned networks into two parts, but the definition of homogeneous part is different from that of Formulation C and D. The homogeneous part of formulation S is in the sense that the VAR coefficients have **identical value**. We refer the result of formulation S as a **similar & differential network**

In general, a joint sparse estimation of $K$ models can be expressed as

$$\underset{A}{\text{minimize}} \quad f(A) + g(A^{(1)}, A^{(2)}, \ldots, A^{(K)}) \tag{11}$$

where $A^{(i)}$ is the parameter of the $i$th model and $A = (A^{(1)}, A^{(2)}, \cdots, A^{(K)})$ is a $K$-tuple of VAR coefficient matrices, the function $f$ is sum-square-error loss, the function $g$ is the structured sparse-inducing regularization term. As a goodness of fit in $\ell_2$-norm sense, the function $f$ can be

$$(1/2)\sum_{k=1}^{K} \|Y^{(k)} - A^{(k)}H^{(k)}\|_F^2. \tag{12}$$

From (4), the Granger causality from time-series $j$ to time-series $i$ is zero if and only if the vector $B_{ij} = \begin{bmatrix}(A_1)_{ij} & \cdots & (A_p)_{ij}\end{bmatrix}$ is zero vector. The group lasso can be employed to regulate the solution with a block sparsity in $B_{ij}$. In our notation, the group lasso penalty is

$$g(A) = \lambda \sum_{i\neq j}\sum_{k=1}^{K} \|B_{ij}^{(k)}\|_2 \tag{13}$$

where we define

$$B_{ij}^{(k)} = [(A_1^{(k)})_{ij} \quad \cdots \quad (A_p^{(k)})_{ij}] \in \mathbf{R}^p.$$

the sparse estimation of $K$ models has been established. When $K = 1$, this formulation is the same as in Haufe's thesis [Hau12], that is a single VAR model estimation with sparse GC matrix as a prior knowledge. We discuss the structured regularization as prior information for each formulation in the following section.

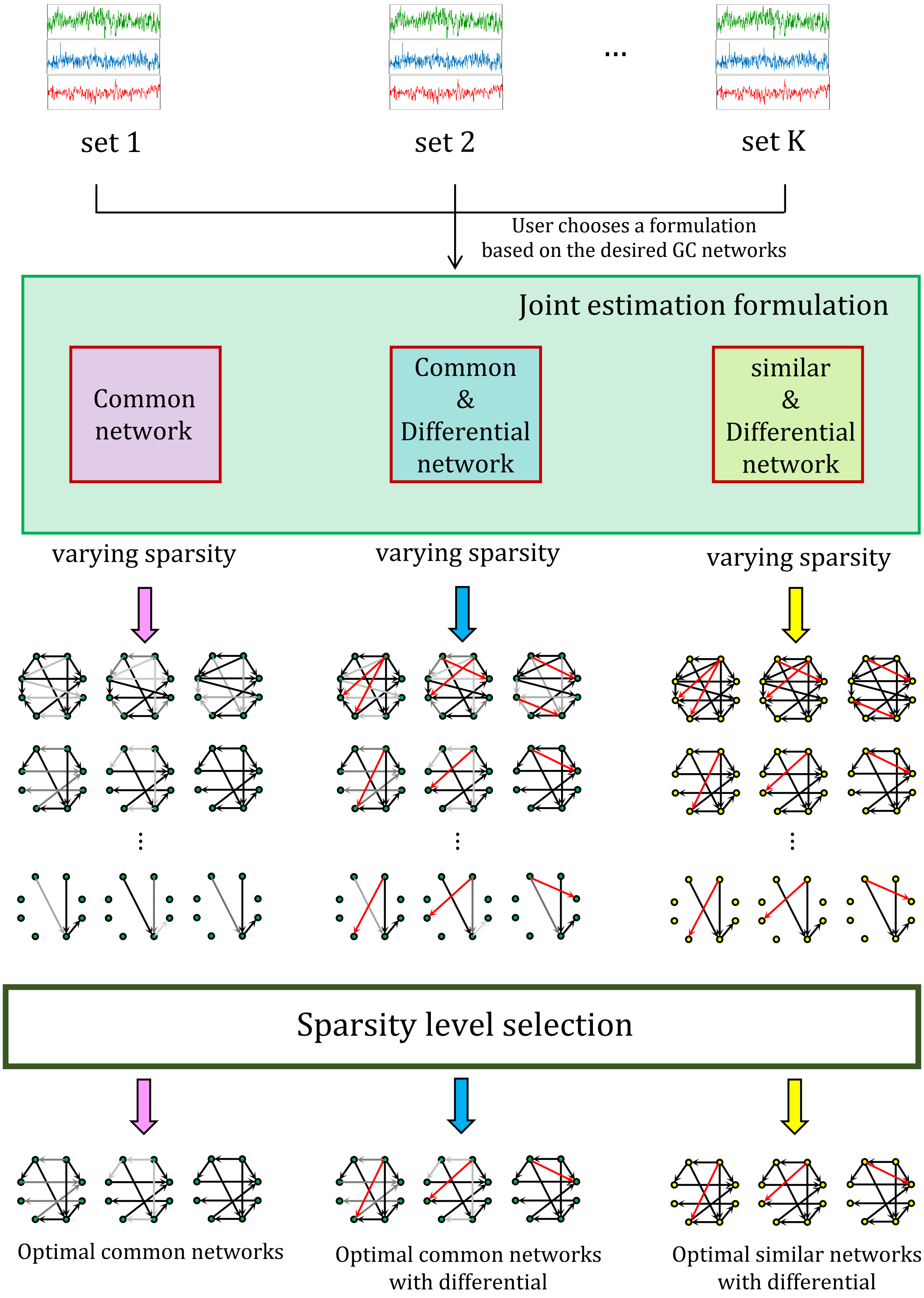

...
set 1
set 2
set K
User chooses a formulation based on the desired GC networks
Joint estimation formulation
Common network
Common & Differential network
similar & Differential network
varying sparsity
varying sparsity
varying sparsity
Sparsity level selection
Optimal common networks
Optimal common networks with differential
Optimal similar networks with differential

Figure 4: Joint Granger graphical models estimation diagram.

## 4.1 Formulations and estimations

The estimation with regularization (13) does not consider any prior information on the relations between models such as similarity or common parts of GC networks. In convex setting, the term $\|B_{ij}^{(k)}\|_2$ is a local regularization, we define this as the differential regularization since it regulates each model differently. To add a notion of global network precisely, we propose two penalty terms to extract all models' common parts. The first penalty is based on the non-convex group norm penalty, which regulates the sparsity pattern of GC matrices to be identical. The second penalty is based on the regularization of block-differences of VAR parameters. The second penalty forces the VAR coefficients to have identical values, making the similarity across all GC matrices.

### 4.1.1 Common estimation of GC networks (Formulation C)

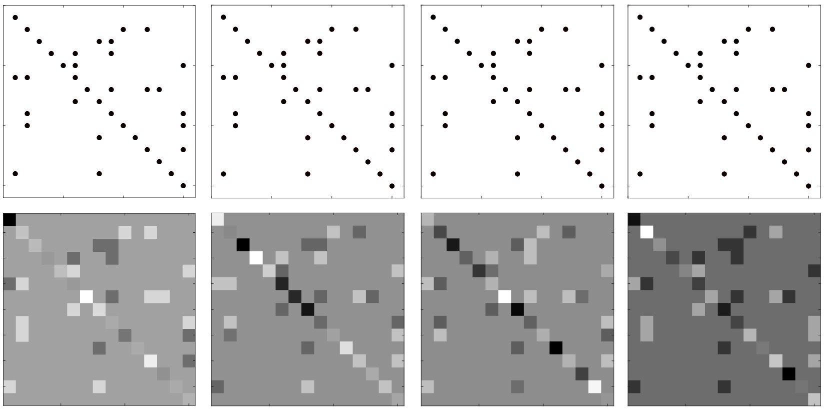


Figure 5: Example of four common Granger networks. (Upper) the sparsity pattern of networks. (Lower) the example of VAR coefficients in each network.

The common sparsity structure can be achieved by regulating a group of variables $B_{ij}^{(k)}, k = 1, 2, \dots, K$ with a group sparse penalty.

We define

$$C_{ij} = \begin{bmatrix} B_{ij}^{(1)} & B_{ij}^{(2)} & \cdots & B_{ij}^{(K)} \end{bmatrix}. \tag{14}$$

As a result from (4), if $C_{ij} = 0$, then all $K$ models have no GC connection from variable $j$ to variable $i$. This suggests us to promote a common sparsity among all $K$ GC matrices by using the regularization term as

$$g(A) = \lambda \sum_{i \neq j} \|C_{ij}\|_2. \tag{15}$$

**Relation to prior work.** This group lasso formulation is an existing estimation method found in [SM19b, GKMM15]. As described in section 3.3, the accuracy of parameter selection depends on the statistical property of group lasso regression. The group lasso can be replaced by a group norm penalty proposed in [HLM$^+$17]. The non-convex formulation of this can be expressed as

$$g(A) = \lambda \sum_{i \neq j} \|C_{ij}\|_p^q. \tag{16}$$

This formulation allows us to determine the Granger network representation among multiple models. The non-zero pattern and sample of VAR coefficients among four Granger graphical models are shown in Figure 5. In the upper part of Figure 5, the nonzero entry $(i, j)$ in GC matrix refers to a Granger connection from variable $j$ to variable $i$. In the lower part of the figure, the intensity of each grid indicates GC strength between variables and directly relates to the magnitude of VAR coefficients.

Although the Granger networks are forced to have an identical pattern, it adds flexibility by allowing each VAR model to have a different value of coefficients. The tuning parameter $\lambda$ directly controls the sparsity of the GC matrix. In group lasso problem, the minimum tuning parameter that the optimal solution is zero or

$$\lambda_{\mathrm{c}} = \inf_{\lambda} \ \{\lambda \mid \hat{A} = 0\} \tag{17}$$

where $\hat{A}$ is obtained from solving (11) with $g$ in (15). The formula is derived in [Son17]. We denote this value of tuning parameter as $\lambda_{\mathrm{c}}$. In the non-convex penalty case, it is known that it should produce a sparser result than the group lasso. With $\lambda = \lambda_{\mathrm{c}}$, the non-convex regularized regression by using penalty in (16) should return the all-zero solution.

### 4.1.2 Common and differential estimation of GC networks (Formulation D)

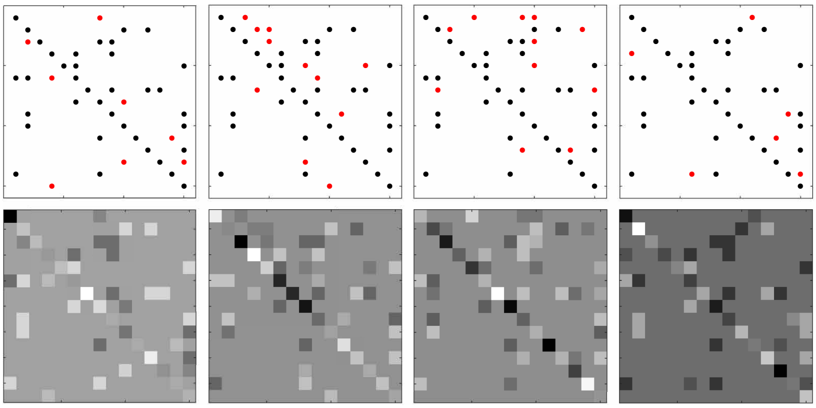


Figure 6: Example of 4 common Granger networks (black dots) with differential pattern (red dots). (Upper) the sparsity pattern of networks. (Lower) the example of VAR coefficients in each network.

The penalty terms (15),(16) rely on an assumption that all multiple time-series have identical Granger network. This assumption may not hold with the presence of heterogeneity of multiple time-series which can be occurred when the data were collected in different scenarios. By adding heterogeneity to the formulation (15), we yield

$$g(A) = \lambda_1 \sum_{i \neq j} \sum_{k=1}^{K} \|B_{ij}^{(k)}\|_2 + \lambda_2 \sum_{i \neq j} \|C_{ij}\|_2 \tag{18}$$

and the non-convex extension is

$$g(A) = \lambda_1 \sum_{i \neq j} \sum_{k=1}^{K} \|B_{ij}^{(k)}\|_p^q + \lambda_2 \sum_{i \neq j} \|C_{ij}\|_p^q \tag{19}$$

This formulation adds more flexibility to the joint estimation by allowing differences in the Granger networks from the first term in (19). The decomposition property of this formulation is visualized in Figure 6. The common part of GC networks in both (15), (18). The sparsity of heterogeneous part and homogeneous part are controlled by varying $\lambda_1$ and $\lambda_2$ respectively. It is worth noting that each group norm penalty term in (18), (19) has different block size. The range of tuning parameter $\lambda_1$ and $\lambda_2$ are also different. We heuristically follow [Son17] to use range of each tuning parameter the same as (15) or set $\lambda_c$ for $\lambda_1$ when fixing $\lambda_2 = 0$ and vice versa.

**Relation to prior works.** In this formulation, we extend the original work of [Son17] by replacing the group lasso penalty to the non-convex group norm penalty. The joint estimation formulation that has the same purpose as this formulation is also presented in [SM19b]. They proposed a two-stage method to estimate the common part and differential part. In the first stage, they employed group lasso regression as same as our formulation C to extract the common part of the GC network. In the second stage, they used estimation residuals from the first stage to estimate the differential part of the Granger network. The two-stage approach for estimation may be optimal in each stage, but may not for both.

### 4.1.3 Similar and differential estimation of GC networks (Formulation S)

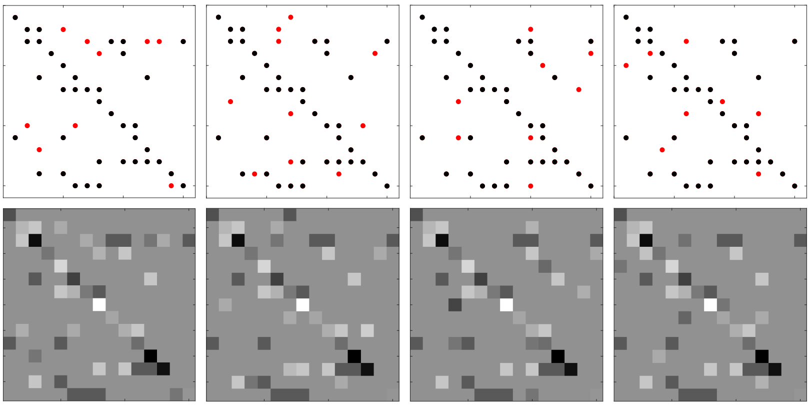


Figure 7: Example of 4 common Granger networks (black dots) with differential pattern (red dots). (Upper) the sparsity pattern of networks. (Lower) the example of VAR coefficients in each network.

In this formulation, we assume that their ground-truth network shared identical value of VAR coefficients and possessed some different part in the network. We propose to extend [Son17] in two different points of view. First, we forced similarity across all models, and replacing the group fused lasso penalty by the non-convex group norm penalty composited with difference matrix. Our extension to the formulation [Son17] is to consider all possible differences of model coefficients. [Son17] proposed a formulation

$$g(A) = \lambda_1 \sum_{i \neq j} \sum_{k=1}^{K} \|B_{ij}^{(k)}\|_2 + \lambda_2 \sum_{k}^{K-1} \sum_{i \neq j} \|B_{ij}^{(k)} - B_{ij}^{(k+1)}\|_2.$$

where only the adjacent model are encouraged to be similar. As an extension to this formulation, we proposed

$$g(A) = \lambda_1 \sum_{i \neq j} \sum_{k=1}^{K} \|B_{ij}^{(k)}\|_2 + \lambda_2 \sum_{k<\ell} \sum_{i \neq j} \|B_{ij}^{(k)} - B_{ij}^{(\ell)}\|_2. \tag{20}$$

which takes difference of all combination into account. As a non-convex extension, we propose

$$g(A) = \lambda_1 \sum_{i \neq j} \sum_{k=1}^{K} \|B_{ij}^{(k)}\|_p^q + \lambda_2 \sum_{k<\ell} \sum_{i \neq j} \|B_{ij}^{(k)} - B_{ij}^{(\ell)}\|_p^q. \tag{21}$$

This formulation promotes both differential sparsity and VAR coefficients similarity in all models. We refer the model learned from this formulation as similar model. It is worth noting that the difference between formulation S and formulation D is in the second term where formulation D permit VAR coefficients to have different value while formulation S force them to be identical. The Figure 7 shows the patterns of four Granger graphical model that the VAR coefficients are identical in the common part.

**Relation to prior works.** The common parts of multiple GC networks can also be described by the similarity of the GC strength. In [Son17], the author proposed to use group fused lasso to regulate the difference between adjacent models. The regularization technique used in [Son17] is limited to the case when similar models are consecutive to each other. This convex regularized VAR model sparse estimation has been proposed in [SM19a], [WBC18]. Recently in [TB20], they proposed joint estimation formulation based on Laplacian regularization. This model has the same interpretation as this formulation. The differences are the type of models and the choice of regulating function. They used Tikhonov regularized the differences between GGM models, which is a smooth regularizer while we used the sparse inducing regularizer.

From all proposed formulation, the vector $C_{ij}$ is an $n(n-1)pK$-dimensional vector rearranged from the VAR coefficients across all lags and all $K$ models. This arrangement complicates the implementation of the algorithm because of the regularized block (14) contains coefficients of all $K$ models. So instead of solving the objective with the coefficients in the fitting term have a different arrangement from the regularization term, we vectorized the fitting term such that its coefficient arrangement matches that of the regularization term.

### 4.1.4 Vectorized model estimation

In this section, we provide our three main formulations in a vectorized format as regularized linear least-square problems so that formulation properties can be discussed and compared with existing works.

In the original problem, we aim to investigate both common and differential GC connections among $K$ sets of multivariate time-series $\{y^{(1)}(t), y^{(2)}(t), \ldots, y^{(K)}(t)\}_{t=1}^{T}$ where $y^{(i)}(t)$ is an $n$-dimensional time-series of $i$th set for $t = 1, 2, \ldots, T$. These time-series are fitted to $p$th order VAR. The vectorized VAR coefficients is the same as $C_{ij}$ in (15). So the vectorized format of formulation (16), (19) and (21), can be expressed as

$$\underset{x}{\text{minimize}} \;\; f(x) + g(x) \tag{22}$$

where $x = [C_{11}, \quad C_{12}, \quad \cdots \quad C_{n,n-1}, \quad C_{n,n}] \in \mathbf{R}^{n^2pK}$ and $f(x)$ is the vectorized loss function given by

$$f(x) = (1/2)\|Gx - b\|_2^2 \tag{23}$$

where $G$ is the matrix size $nTK \times n^2pK$. The regularization terms are

- **Vectorized formulation** (15), (16)

$$g(x) = \lambda \|\mathcal{P}x\|_{p,q}^{(pK)} \tag{24}$$

- **Vectorized formulation** (18), (19)

$$g(x) = \lambda_1 \|\mathcal{P}x\|_{p,q}^{(p)} + \lambda_2 \|\mathcal{P}x\|_{p,q}^{(pK)} \tag{25}$$

- **Vectorized formulation** (20), (21)

$$g(x) = \lambda_1 \|\mathcal{P}x\|_{p,q}^{(p)} + \lambda_2 \|\mathcal{D}x\|_{p,q}^{(p)} \tag{26}$$

where $\mathcal{P}$ is a projection matrix and $\mathcal{D}$ is a difference matrix. These notions enable us to insert prior knowledge on which links of GC or their differences are taken into account. If so, the parameters involved in those GC links or their differences will be included in the mapped coordinate to be regularized. In the case different from ours, the matrix $\mathcal{P}, \mathcal{D}, G$ can be any matrix; however, it is worth noting that the convergence and efficiency of numerical methods to solve these problems depends on the structure of matrix $G, \mathcal{P}, \mathcal{D}$.

In our setting, the matrix $G$ is a tall matrix if $T > np$ and $G^T G$ has a block-diagonal structure. The matrix $\mathcal{P}$ has full row rank. The matrix $\mathcal{D}$ is a tall matrix, so it cannot be full row rank. The properties of these matrices will affect the algorithm. We discuss this in the algorithm section. The detailed explanation of projection matrix and difference matrix are given in the Appendix A. In each of our formulation, the density of heterogeneous part and homogeneous part in the Granger network can be controlled by varying $\lambda_1$ and $\lambda_2$, respectively. The formulation is a special case of formulation D when $\lambda_1 = 0$. The estimated VAR models in each varying pair of $\lambda_1, \lambda_2$, are stored and prepared for the network density selection process.

## 4.2 GC network density selection

The performance of any regularized regression depends on the selection of tuning parameters methods or the model selection. The tuning parameters that affect model complexity are $\lambda_1, \lambda_2$, and VAR order $p$. The first two parameters control sparsity directly, but the tuning parameter $p$ controls only the number of model parameters. In our work, we assume that $p$ can be fixed, so we consider only sparsity level selection. The model selection criteria must be introduced to select the best pair of $\lambda_1, \lambda_2$ for an optimal sparsity level. Due to the bias-variance decomposition principle, the trade-off between complexity and model quality is inevitable. The regularization level that yields an optimal trade-off between the fitting and complexity will be selected. To select the model selection criteria, we split the types of selection into two types, the one that relies on the trade-off between fitting and model complexity and relies on the sub-sampling technique.

For the first type, we consider AIC (Akaike Information Criteria), BIC (Bayesian Information Criteria) which are defined as

$$\mathrm{AIC}(\lambda_1, \lambda_2) = -2\mathcal{L}(\lambda_1, \lambda_2) + 2\mathrm{df}(\lambda_1, \lambda_2) \tag{27}$$
$$\mathrm{BIC}(\lambda_1, \lambda_2) = -2\mathcal{L}(\lambda_1, \lambda_2) + \mathrm{df}(\lambda_1, \lambda_2) \log(T) \tag{28}$$

where $\mathcal{L}$ is the log-likelihood of the model, $T$ is number of time points and $\mathrm{df}$ is the measure of complexity of the models or the degree of freedom. However, the asymptotic properties of these criteria depend on the choices of the degree of freedom [HTF01]. We heuristically selected the degree of freedom as the number of all nonzero parameters which is the same as lasso regression. In [Lüt05], the log-likelihood function of VAR is stated as

$$\mathcal{L} = -(nT/2)\log 2\pi - (T/2)\log\det\hat{\Sigma} - (1/2)\mathrm{tr}[(Y - \hat{A}H)\Sigma^{-1}(Y - \hat{A}H)^T] \qquad (29)$$

where $\hat{\Sigma} = \frac{1}{T-p}EE^T$ with $E = Y - \hat{A}H$, $\hat{A}$ is a sparsity constrained least-square which the sparsity is learned from the estimation process. By plugging $\hat{\Sigma}$ into (29), we obtain

$$\mathcal{L} = -(nT/2)\log 2\pi - (T/2)\log\det\hat{\Sigma} - (n/2)(T-p). \qquad (30)$$

we select $(\lambda_1, \lambda_2)$ as,

$$(\hat{\lambda}_1, \hat{\lambda}_2)_{\mathrm{AIC}} = \underset{\lambda_1,\lambda_2}{\text{argmin}} \quad \mathrm{AIC}(\lambda_1, \lambda_2)$$
$$(\hat{\lambda}_1, \hat{\lambda}_2)_{\mathrm{BIC}} = \underset{\lambda_1,\lambda_2}{\text{argmin}} \quad \mathrm{BIC}(\lambda_1, \lambda_2)$$

For the second type of the model selection criteria, we consider the $K$-fold cross-validation (CV). As illustrated in Figure 8, $K$-fold CV splits the data into $K$ chunks. One chunk is selected as a validation data set and the remaining data are the training set. The training data is used to estimate the model and evaluate error on the validation set. This process is repeated $K$ times to average error on validation set as described in figure 8. In the $i$th iteration of CV, the sum squared error (SSE) of model with $(\lambda_1, \lambda_2)$ is given by $\mathrm{SSE}(\lambda_1, \lambda_2, i) = \|(y_{\mathrm{validate}} - \hat{y}_{\mathrm{train}})_i\|_2^2$. This method selects the model such that,

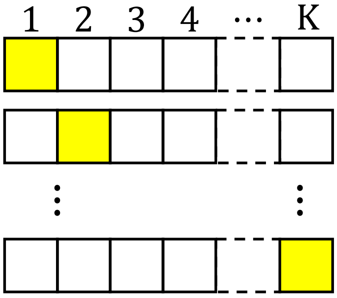


Figure 8: Visualization of $K$-fold cross-validation method.

$$(\hat{\lambda}_1, \hat{\lambda}_2)_{\mathrm{CV}} = \underset{\lambda_1,\lambda_2}{\text{argmin}} \quad (1/K)\sum_{i=1}^{K}\mathrm{SSE}(\lambda_1, \lambda_2, i)$$

or, in other words, the model that yields the lowest averaged validation error will be selected. In our settings, the sparse solutions are expected but the goal of cross-validation is to minimize the sum-squared-error in the validation set, so this method was shown to typically provide dense solution. This method simply over-selects the variables in the model.

In a good variable selection method, the insignificant variables should be crossed out in the process, both estimation and model selection. The stability selection [MB10] extends this paradigm by introducing the stability of estimated variables. A stable variable is a variable that rarely estimated as zero by a sparse estimation method when the data changed. The stability selection changes the data by sub-sampling the data and estimate the model for a given amount of regularization. In a sparse linear regression context, the sub-samples are drawn by half of the sample size. Each sub-sample is used as data of the sparse estimation and repeats the process with the replacement of drawn data multiple times. A threshold of occurrence times to regard a variable as a stable predictor is another tuning parameter to be manually selected. Unlike BIC or AIC, this method is not a direct way for regularization level selection. The stability selection is rather an ensemble method that is used to select stable variables over a given range of regularization. In our case, the information-theoretic criterion is more useful in our case than the stability selection method because of two reasons. First, the stability selection requires sub-sampling of data, which is not practical for high dimensional model estimation. Second, the sequence of time-series data is strongly correlated with the past of itself. This indicates that the time-series cannot be randomly sub-sampled, so the time-series must be sub-sampled in a block-wise manner, and the number of blocks must be high enough for an accurate, stable solution selected by this method.

The computation expenses for each selection technique can be estimated. The penalties $(\lambda_1, \lambda_2)$ are varied in 2D grid sized $L \times L$. In AIC and BIC, the value of criteria can be directly assigned to each model. As a result, there are $L^2$ models to be estimated in both AIC and BIC. In $K$-fold CV, each model has to be estimated $K$ times, resulting in the $KL^2$ models to be determined. Similar to the $K$-fold CV, each model has to be estimated $I$ times, resulting in the $IL^2$ models to be determined. For these reasons, the computational complexity of both AIC, BIC is much lower than the sub-sampling methods. Since we seek for simpler models, the BIC score is preferred over AIC because it tends to choose a sparser model.

At this point, the problem statements and model selection techniques were concisely stated. However, the formulations (15), (18), (20) are non-smooth convex optimization problems and (16), (19), (21) are non-smooth non-convex optimization problems. The non-smoothness brings us difficulties in the sense that the gradient method is not available, and the non-convex optimization is more difficult to compute than the convex optimization, and the available solvers are limited. In the next section, we discuss suitable algorithms to solve these formulations in a large scale setting.

# 5 Algorithms

In all of our formulations, the optimization can be expressed, as stated in (22). For explicit description, we state the problem again as

$$\underset{x}{\text{minimize}} \; f(x) + g(x)$$

where $f$ is a sum-squared error loss function which is convex and has a Lipschitz continuous gradient. The term $g$ depends on formulation parameters $p, q$. If $p = 2$ and $q = 1$ then $g$ is convex and not differentiable at zero as its gradient is not defined. In our non-convex formulation, the term $g$ is non-convex and not differentiable at zero. The gradient-based algorithm cannot be applied since the solution of the formulation is encouraged to be concentrated at zero, where the gradient is not defined. The problem (22) and its properties can be arranged into a format to which existing proximal algorithms can be applied. In the subsequent section, we summarized some background on proximal algorithms from [PB14].

## 5.1 Proximal operator

The definition of proximal operator is

$$\mathbf{prox}_{\lambda h}(v) = \underset{x}{\text{argmin}} \; h(x) + \frac{1}{2\lambda}\|x - v\|^2 \tag{31}$$

where $h$ is convex function, $x, v \in \mathbf{R}^n, \lambda > 0$. It can directly seen that if $f$ is separable in $x$, or $h(x) = \sum_i r_i(x_i)$ and $x_i$ is a sub-block of $x$, then its proximal operator is separable in each $x_i$ [PB14]. This property allows the parallel processing of proximal operator evaluation. For example, the proximal operator of $h$ is

$$\begin{aligned}
\mathbf{prox}_{\lambda h}(v) &= \underset{x}{\text{argmin}} \; \sum_{i=1}^{K} r_i(x_i) + (1/2\lambda)\|x - v\|_2^2 \\
\mathbf{prox}_{\lambda r_i}(v) &= \underset{x}{\text{argmin}} \; r_i(x_i) + (1/2\lambda)\|x_i - v_i\|_2^2 \\
\mathbf{prox}_{\lambda h}(v) &= (\mathbf{prox}_{\lambda r_1}(v_1), \mathbf{prox}_{\lambda r_2}(v_2), \cdots, \mathbf{prox}_{\lambda r_K}(v_K))
\end{aligned}$$

In our formulations, the group norm penalties are also separable.

## 5.2 Proximal algorithms

The proximal algorithms are based on proximal operator. The proximal operator can be interpreted in many ways. The most natural way to us is the interpretation as gradient flow system. Consider the system

$$\dot{x}(t) = -\nabla f(x(t)) \tag{32}$$

where the stationary point of the system is the local optima of $f$. The discretization methods of this system can be made as

$$\frac{x_{k+1} - x_k}{\lambda} = -\nabla f(x_k) \tag{33}$$

$$\frac{x_{k+1} - x_k}{\lambda} = -\nabla f(x_{k+1}) \tag{34}$$

where (33) represents a Forward-Euler discretization and (34) represents a Backward-Euler discretization. The Forward-Euler one can be rearranged into the well-known gradient descent algorithm. The equation (34) can be replaced using the definition of (31) as

$$x_{k+1} = \mathbf{prox}_{\lambda f}(x_k)$$

which is called the **proximal point** algorithm. For a typical gradient method, a closed-form gradient must be accessible and easy to evaluate. If not, the gradient must be estimated. The inexactness of gradient estimation will affect the convergence of algorithm. Similarly, to gain a benefit of proximal algorithms, the proximal operator should be in a closed-form expression. However, many problems, such as lasso problem, its proximal operator has a closed-form expression for each term but not for both. This problem indicated the proximal point algorithm is not able to solve this problem efficiently. In the sparse estimation problem, the optimization problem is in the format of

$$\underset{x}{\text{minimize}}\ f(x) + g(x) \tag{35}$$

where the gradient of $f$ is Lipschitz continuous and the gradient of $g$ is undefined at $x = 0$. When dealing with a gradient discontinuity for a convex function, the notion of subgradients must be introduced. For a convex $g$, any vector $s$ that satisfied the inequality,

$$g(x) \geq g(z) + s^T(x - z),$$

is called a subgradient of $g$ at $x = z$ denoted as $\partial g(z)$. By considering the system(32) and using notion of subgradient, the new subgradient flow system is

$$\dot{x}(t) = -\nabla f(x(t)) - \partial g(x(t)) \tag{36}$$

The terms related to $x_k$ can be rearranged on the same side of the discretization as in (33) and the terms related to $x_{k+1}$ can also be arranged on the other side of the discretization as in (34). By combining forward and backward discretization, we achieve

$$x_{k+1} + \lambda \partial g(x_{k+1}) = x_k - \lambda \nabla f(x_k) \tag{37}$$

With notion of proximal operator, the equation (37) reduces to

$$x_{k+1} = \mathbf{prox}_{\lambda g}(x_k - \lambda \nabla f(x_k)) \tag{38}$$

This algorithm also known as **proximal gradient method** or forward-backward algorithm as its discretization interpretation.

In some kind of problems, a proximal operator of a non-smooth function is easy to evaluate but not when its argument is a composition with a linear transformation $L$. For example,

$$\underset{x}{\text{minimize}}\ \ f(x) + g(Lx). \tag{39}$$

This problem can be transformed into constrained format which can be solved by the well-known ADMM (Alternating Direction Method of Multipliers) algorithm. We will refer this algorithm as vanilla ADMM [BPC$^+$11]. ADMM was originally used to solve convex problems that in the format of

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array} \tag{40}$$

This algorithm is a variant of augmented Lagrangian methods and its augmented Lagrangian is

$$\mathcal{L}(x, z, \lambda, \rho) = f(x) + g(z) + y^T r + (\rho/2)\|r\|_2^2 \tag{41}$$

where $x, z$ are both primal variables but $z$ can be referred as the splitting variable, and $r = Ax + Bz - c$ is a primal residuals. The variable $y$ is the dual variable. The algorithm parameter $\rho$ is a penalty parameter which plays important role in convergence speed of convex problem and convergence in our non-convex formulation . The vanilla ADMM [BPC+11] simply updates primal variable $x$ and $z$ that minimize the augmented Lagrangian (41) in the alternating scheme and then update the dual variable $y$. The algorithm updates are as follow.

**Algorithm 1:** Vanilla ADMM [BPC+11]

```
initialization: $x, z,\ \rho > 0.$;
while $\|r\|_2 \geq \epsilon_{\text{pri}}$ and $\|s\|_2 \geq \epsilon_{\text{dual}}$ do
    $x^+ = \text{argmin}_x\, f(x) + \frac{\rho}{2}\|Ax + Bz - c + \frac{y}{\rho}\|_2^2,$ ;
    $z^+ = \text{argmin}_z\, g(z) + \frac{\rho}{2}\|Ax^+ + Bz - c + \frac{y}{\rho}\|_2^2,$ ;
    $y^+ = y + \rho(Ax^+ + Bz^+ - c),$
```

The primal and dual residuals are $\|r^+\|_2 = \|Ax^+ + Bz^+ - c\|_2$, $\|s^+\|_2 = \rho\|A^T B(z^+ - z_k)\|_2$ respectively. The optimality condition can be checked through the convergence of these measures. We used $\epsilon_{\text{pri}}, \epsilon_{\text{dual}}$ as computed in [BPC+11]. When employing vanilla ADMM to solve joint estimation, the problem with penalty $g$ in (25) can be converted to ADMM format as

$$\begin{array}{ll} \text{minimize} & \|Gx - b\|_2^2 + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

with problem parameters

$$A = \begin{bmatrix} \mathcal{P} \\ \mathcal{P} \end{bmatrix}, B = -I, c = 0 \tag{42}$$

and the problem with penalty $g$ in (26), the algorithm parameters $A, B, c$ in (42) changed to

$$A = \begin{bmatrix} \mathcal{P} \\ \mathcal{D} \end{bmatrix}, B = -I, c = 0 \tag{43}$$

These settings allow exploitation in the ADMM update steps. With the sum-squared loss term $\|Gx - b\|_2^2$, the $x$ update step reduces to solving a system of linear equation instead. Moreover, when $B = -I$, the form in $z$-update step is the same as the proximal operator of $g/\rho$ evaluated at $Ax^+ + y/\rho$. Both steps in vanilla ADMM reduce to the form,

$$\begin{aligned} x^+ &= \{x \mid (\rho A^T A + G^T G)x = G^T b + \rho A^T(y - z)\} \\ z^+ &= \mathbf{prox}_{(g/\rho)}(Ax^+ + y/\rho). \end{aligned}$$

If the proximal operator has a closed-form expression, the significant computational cost only depends on solving linear equation. The linear equation in the new $x$ update step has block-diagonal structure which can be exploited further to reduce the computational cost. However, the vanilla ADMM only has theoretical global convergence for our convex problems. The important question is *Do these algorithms globally converge in the non-convex setting as well?* In the following part, we discuss about how we employ ADMM algorithm to solve the joint estimation with convex penalties first and then discuss the algorithm that are available to solve our problems with non-convex penalties in the last part.

## 5.3 Choices of algorithms

The main problem (22) and with different choices of penalties (24), (25), (26) can be represented in a general form as

$$\underset{x}{\text{minimize}} \;\; f(x) + \lambda_1 h_1(L_1 x) + \lambda_2 h_2(L_2 x) \tag{44}$$

This is not readily in the same format as vanilla ADMM except when regularization term is (24).

The regularization term in (44) is compactly defined as

$$g(x) = \lambda_1 h_m(x) + \lambda_2 h_n(x) \tag{45}$$

where $h_m(x) = \|x\|_{p,q}^{(m)} = \sum_{i=1}^{K} \|x_i\|_p^q$ which is a group norm penalty with parameter $(p, q, m)$ with $x = (x_1, x_2, \ldots, x_K) \in \mathbf{R}^{mK}, x_i \in \mathbf{R}^m$. The superscript $m$ is used to point out that the vector $x$ can be partitioned with a certain block size, which is $m$ in this case, and the norm is evaluated on the sub-block of that size.

To convert the general form (44) to ADMM format, we split the variable as $z = (z_1, z_2) = Lx = (L_1 x, L_2 x)$. so by using the property of proximal operator (5.1), the proximal operator of function (45) will be

$$\textbf{prox}_{\lambda g}(v) = \begin{bmatrix} \textbf{prox}_{\lambda_1 h_m}(v_1) \\ \textbf{prox}_{\lambda_1 h_n}(v_2) \end{bmatrix}$$

with $v = (v_1, v_2)$. This expression in each formulation has a different form. For $p = 2, q = 1$, the $i$th block of its proximal operator when the block size is $m$ is

$$(\textbf{prox}_{\lambda h_m}(v))_i = \max\{0, 1 - \frac{\lambda}{\|v_i\|_2}\}$$

The proximal operators are not always in closed-form for all of the pairs $(p, q)$ with $q < 1$. For example, the single variable optimization problem

$$\underset{x}{\text{minimize}} \;\; \lambda |x|^{0.1} + (ax - b)^2$$

when $x > 0$, the critical point can be obtained from solving the zero gradient condition,

$$(1/10)\lambda + 2a^2 y^{19} - 2aby^9 = 0$$

for $y$, with $y = x^{1/10}$. This problem is impossible to solve analytically as a direct consequent from Abel's theorem [DF03], which stated that there is no closed-form expression of the roots of fifth-degree polynomials or higher. In [HLM+17], they derived closed-form of proximal operators of group norm penalty for some pairs of $(p, q)$.

We selected the value of $p = 2, q = 1/2$ due to the fact that its proximal operator has a closed-form expression and the experimental result in [HLM+17] yielded that $q = 1/2$ is the best among all $0 \leq q \leq 1$. For this pair of $p, q$, the closed-form expression of $i$th block of proximal operator when block size is $m$ is

$$(\textbf{prox}_{\lambda h_m}(v))_i = \begin{cases} \left( \frac{16\|v_i\|_2^{3/2} \cos^3(R(v_i))}{3\sqrt{3}\lambda + 16\|v_i\|_2^{3/2} \cos^3(R(v_i))} \right) v_i, & \text{if } \|v_i\|_2 > (3/2)\lambda^{2/3} \\ \mathbf{0}, & \text{if } \|v_i\|_2 \leq (3/2)\lambda^{2/3} \end{cases} \tag{46}$$

where $R(x) = \pi/3 - \arccos(\frac{\lambda}{4}(\frac{3}{\|x\|_2})^{3/2})$.

In this thesis, the formulations are based on non-convex penalties; however, the convergence of our convex cases still need to be clarified. We split the content of this section into two parts. The detail of solving convex formulation and their convergence property is given in the first part. The second part contains the literature review on non-convex and non-smooth optimization in the sparse regression problems.

### 5.3.1 Convex penalty

In our convex cases, the theoretical convergence of ADMM algorithm is global which follows from the convergence analysis of [BPC⁺11]. The problem (44) is in ADMM format when $A = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}$, $B = -I$, the ADMM algorithm for solving (24)-(26) can be explicitly described as the following,

```
Algorithm 2: SparseGrangerNet
initialization: $x, z = (z_1, z_2), y = (y_1, y_2),\ \rho > 0.$;
while $\|r\|_2 \geq \epsilon_{\text{pri}}$ and $\|s\|_2 \geq \epsilon_{\text{dual}}$ do
    $x^+ = \left(\rho(L_1^T L_1 + L_2^T L_2) + G^T G\right)^{-1}(G^T b + \rho \begin{bmatrix} L_1^T & L_2^T \end{bmatrix}(y - z))$ ;
    $z_1^+ = \mathbf{prox}_{(\lambda_1 h_{n_1}/\rho)}(L_1 x^+ + y_1/\rho)$ ;
    $z_2^+ = \mathbf{prox}_{(\lambda_2 h_{n_2}/\rho)}(L_2 x^+ + y_2/\rho)$ ;
    $y_1^+ = y_1 + \rho(L_1 x^+ - z_1^+)$;
    $y_2^+ = y_2 + \rho(L_2 x^+ - z_2^+)$
```

where $(L_1, L_2) = (\mathcal{P}, \mathcal{P})$ and $(\mathcal{P}, \mathcal{D})$ in the formulation (25), (26) respectively. The formulation (24) is solved as in (25) with $\lambda_1 = 0$. The value of $(n_1, n_2)$ is $(p, pK)$ in the formulation (24), (25) but $(p, p)$ in the formulation (26) where $p$ is the number of lag and $K$ is number of models. This technique is not new, the SDMM algorithm is similar to our algorithm but has been derived in the different point of view on the objective function. The algorithm (2) can be extended to solve the problems in the form of

$$\underset{x}{\text{minimize}} \;\; f_0(x) + f_1(L_1 x) + \cdots + f_m(L_m x)$$

which the primal, non-splitting variable $x$ presented in the cost function explicitly. The SDMM algorithm [CP11] is used to solve problems in the form of

$$\underset{x}{\text{minimize}} \;\; f_1(L_1 x) + f_2(L_2 x) + \cdots + f_m(L_m x)$$

which only primal, splitting variables ($z_i = L_i x$) are in the cost function. One can understand that solving linear equation in $x$-update step is inevitable. In SDMM, the $x$-update step takes the form of

$$\begin{aligned} x^+ &= \underset{x}{\text{argmin}} \; \frac{\rho}{2}\left\| Qx - z + \frac{y}{\rho} \right\|_2^2 \\ x^+ &= \left\{ x \mid Q^T Q x = Q^T(z - \frac{y}{\rho}) \right\} \end{aligned}$$

where $Q = (G, L_1, L_2, \ldots, L_m)$ which is a row concatenation of these matrices. This confirmed the necessity of solving system of linear equations for both SDMM and ours in the $x$ update step. However, the way SDMM algorithm plug the loss function into the update of splitting variables causes the additional step of solving system of linear equations due to the sum-squared loss term. The variable $z_1$ update step in SDMM, that applied to our problem with $L_1 = G$, take the form of,

$$z_1^+ = \underset{z_1}{\text{argmin}} \; \|z_1 - b\|_2^2 + (\rho/2)\|Gx^+ - z_1 + (y_1/\rho)\|_2^2.$$

This extra step involved the solution of the system of linear equations. With this reason, computational complexity of SparseGrangerNet is computationally cheaper than SDMM.

### 5.3.2 Non-convex penalty

As a remark, the general form of our formulations (24),(25) ,(26) is

$$\underset{x}{\text{minimize}}\ f(x) + \lambda_1 h_m(L_1 x) + \lambda_2 h_n(L_2 x) \coloneqq f(x) + g(x) \tag{47}$$

where $f(x)$ is a smooth loss function, $h_m, h_n$ are the group norm penalty with block size $m, n$ respectively. When employing this formulation to jointly estimate multiple Granger graphical models, there are two cases of $L_i$. The first case is when $L = \mathcal{P}$ or a projection operator. This matrix is fat matrix with full row rank. The second case is when $L = \mathcal{D}$ or the difference matrix. In our models, the difference matrix is a tall matrix as we intend to penalize difference of all combination of the models. The detailed explanation of these notions are presented in the section 4.1.4. As shown in (42), (43), these problems can be solved by ADMM algorithm.

**ADMM** Variants of ADMM can be applied if they have splitting property to deal with the linear transformation. The variants of ADMM that are in our interest are the Bregman ADMM and the multi-block ADMM. The Bregman-ADMM is proposed in [WB14] for replacing the Euclidean distance in the ADMM update step with the Bregman distance. This only reduced the computational complexity of ADMM. In multi-block ADMM, the constraint is $\sum_{i=1}^N A_i x_i = c$. This definition collides with the vanilla ADMM when $N = 2$. However, it is known that even in the convex-case, this variant of ADMM may diverge [CHY16]. The convergence analysis of splitting algorithm in literature has different settings from our setting in (47) [HLR16], [OCBP14]. Even when the problem is the same as ours, most of them restricted the matrix $A$ in (40) to be full row rank [GHW17], [LP15], [WCX18], [ZQG16], [ST19]. The rank assumption was assumed in two contexts. First, the full row rank assumption is used for bounding the augmented Lagrangian to be a strictly decreasing sequence generated from the ADMM update step [LP15], [WCX18], [ZQG16]. In the second context, the full row rank assumption was used to guarantee the existence of critical points of the problem [ST19]. To be more specific in the second case, the problem

$$\underset{x}{\text{minimize}}\, f(x) + g(Ax) \tag{48}$$

has the first-order optimality condition as

$$0 \in \nabla f(x) + A^T \partial g(Ax). \tag{49}$$

The optimality condition is sufficiently satisfied when the matrix $A$ is full row rank or a surjective mapping [ST19]. However, the full row rank may be too conservative because the $-\nabla f(x)$ may lie in the row space of $A$. Some literature also provided a scheme that does not assume the surjective mapping assumption, such as in semi convex setting [MSMC15], [ZS19]. In our setting, the weakest assumption for the theoretical global convergence of vanilla ADMM has been proposed in [WYZ19]. They investigated the convergence of the multi-block ADMM algorithm. One of their sufficient conditions is $\mathbf{range}(B) \subset \mathbf{range}(A)$. As shown in (42), (43), our matrix $B$ is a negative identity matrix and the matrix $A$ is a row-concatenation of two matrices. With the concatenation, the row space of $B$ never be a proper subset of row space of $A$. Our non-convex formulations directly violate the assumption in their convergence analysis. This restriction is used to bound the augmented Lagrangian to force it to be a strictly decreasing sequence.

**Other proximal gradient methods** In non-convex formulation (16), the linear transformation $\mathcal{P}$ is a projection operator. The proximal operator of the term that composite with a projection matrix is easy to compute, so the proximal gradient methods can be used to solve this problem efficiently by virtue of the separability property of the proximal operator. The global convergence of the proximal gradient methods in this non-convex problem is derived under Kurdyka-Łojasiewicz (KL) framework due to its descent property [HLM$^+$17]. In [XCXZ12], [ZMZ$^+$13] they proposed to use proximal gradient to solve $\ell_{1/2}$ (iterative half-thresholding algorithm), $\ell_q$ (Generalized iterated shrinkage algorithm, GISA) norm penalization respectively. These are special cases of [HLM$^+$17]. Moreover, [ZMZ$^+$13] proposed an algorithm to evaluate the proximal operator of $\ell_q$ penalty. They derived the threshold bound of the proximal operator to determine whether the variable is zero or not first. If not, they employed the Gauss-Seidel algorithm to solve a nonlinear equation as an evaluation of the proximal operator. This is possible because of the separation of variables in the proximal operator of $\ell_q$ penalty. In the convex case, the proximal gradient methods have their accelerated version called accelerated proximal gradient method (APG). An example of this algorithm is known as FISTA [BT09]. In convex problems, the error bound as a function of iteration $k$ of APG is in order of $\mathcal{O}(1/k^2)$ while the proximal gradient algorithm only has the iteration complexity in order of $\mathcal{O}(1/k)$ [TY10]. However, the APG algorithm does not generate decreasing sequences of objective value as in the proximal gradient algorithm. Like the ADMM case, the strictly decreasing sequence of Lagrangian in (47) is one of sufficient condition to conclude a global convergence of iterations [ABS13]. In [LL15], they proposed variants of APG, which denotes as monotone APG algorithm and non-monotone APG algorithm to solve the non-monotone sequence generated by the APG. In monotone APG, they add a monitoring step that detects the descent of the loss function on an accelerated update step. If the step makes the loss increase, the proximal gradient step that is a descent step is used instead. However, they stated that this algorithm is too conservative and maybe too slow. They further extended this into non-monotone APG (nmAPG). The convergence proof in the KL framework requires the descent property of the objective function. The global convergence of nmAPG is achieved from forcing the decaying weighted average of the objective function to be a strictly decreasing sequence. This adds flexibility to the iteration update. The idea behind this is to use the monitoring step as same as mAPG. The nmAPG algorithm is shown in Algorithm 3.

**Line search for step size selection** In order to have a global convergence, the step size of nmAPG must be less than the reciprocal of Lipschitz constant of the $\nabla f(x)$. We followed the original literature to used Barzilai-Borwein (BB) line search and backtracking to achieve a larger step size. The idea behind BB line search is to estimate the curvature of the update step by choosing step size $\alpha$ such that $\alpha I$ is the best estimation of Hessian. If the line search criteria does not satisfied, the step size is scaled down with a factor of $\rho$. The line search rule is described in (4), (5).

With backtracking rule (4), (5), there would be no prior knowledge on Lipschitz constant. However, if the choice of backtracking parameter $\rho$ is poorly chosen, the algorithm may converge slowly. This can be prevented by setting up a safeguard step to select a larger step size between one that yielded from backtracking line search and one that yielded from the reciprocal of Lipschitz constant. Even though the variants of proximal gradient methods are proved to converge in our non-convex formulation (16), these algorithms cannot be applied to formulation $D, S$. The closed-form proximal operator of the penalty terms must be available for large scale computation otherwise its numerical solution should be sufficiently cheap. The convergence of

**Algorithm 3:** Non-monotone APG [LL15]

```
Input: $x_0, G, b$
Output: $\hat{x}$
Initialize: $z = x = x^-, t = 1, t^- = 0, \eta \in [0, 1), \delta > 0, q = 1,$
$\alpha_x \leq 1/\|G\|_2, \alpha_y \leq 1/\|G\|_2.$
while $\|x^+ - x\|_2 \geq \epsilon$ do
    $y = x + \frac{t^-}{t}(z - x) + \frac{t^- - 1}{t}(x - x^-),$
    $z^+ = \mathbf{prox}_{\alpha_y g}(y - \alpha_y G^T(Gy - b)),$ [Can be replaced with Subroutine 4]
    if $F(z^+) \leq c - \delta\|z^+ - y\|_2^2$ then
        $x^+ = z^+$
    else
        $v^+ = \mathbf{prox}_{\alpha_x g}(x - \alpha_x G^T(Gx - b)),$ [Can be replaced with Subroutine 5]
        $x^+ = \begin{cases} z^+, \textbf{if } F(z^+) \leq F(v^+), \\ v^+, \textbf{else}, \end{cases}$ ,
        $t^+ = (1/2)(\sqrt{4t^2 + 1} + 1),$
        $q^+ = \eta q + 1,$
        $c^+ = (\eta q c + F(x^+))(q^+)^{-1}.$
where $F(x) = (1/2)\|Gx - b\|_2^2 + g(x)$ in (24).
```

**Subroutine 4:** Barzilai-Borwein Backtracking line search for $\alpha_y$ [CP11]

```
Input: $G, y, y^-, \rho$
Output: $z^+$
Initialize: $s = y - y^-, r = G^T G s, \alpha_y = \frac{\|s\|_2^2}{s^T r}$, $0 < \rho < 1$
while $F(z^+) \geq F(y) - \delta\|z^+ - y\|_2^2,$ and $F(z^+) \geq c - \delta\|z^+ - y\|_2^2,$ do
    $z^+ = \mathbf{prox}_{\alpha_y g}(y - \alpha_y G^T(Gy - b)),$
    $\alpha_y = \rho\alpha_y,$
where $F(x) = (1/2)\|Gx - b\|_2^2 + g(x)$ with $g(x)$ in (24).
```

**Subroutine 5:** Barzilai-Borwein Backtracking line search for $\alpha_x$ [CP11]

```
Input: $G, x, y^-, \rho$
Output: $v^+$
Initialize: $s = x - y^-, r = G^T G s, \alpha_x = \frac{\|s\|_2^2}{s^T r}$, $0 < \rho < 1$
while $F(v^+) \geq c - \delta\|v^+ - x\|_2^2$ do
    $v^+ = \mathbf{prox}_{\alpha_x g}(x - \alpha_x G^T(Gx - b)),$
    $\alpha_x = \rho\alpha_x,$
where $F(x) = (1/2)\|Gx - b\|_2^2 + g(x)$ with $g(x)$ in (24).
```

|  | Formulation | Algorithms | Convergence proof |
|---|---|---|---|
| Convex | $\min_x (1/2)\|Gx-b\|_2^2 + \lambda\|\mathcal{P}x\|_{2,1}^{(pK)}$ | ADMM | [BPC+11] |
|  | $\min_x (1/2)\|Gx-b\|_2^2 + \lambda_1\|\mathcal{P}x\|_{2,1}^{(p)} + \lambda_2\|\mathcal{P}x\|_{2,1}^{(pK)}$ | ADMM | [BPC+11] |
|  | $\min_x (1/2)\|Gx-b\|_2^2 + \lambda_1\|\mathcal{P}x\|_{2,1}^{(p)} + \lambda_2\|\mathcal{D}x\|_{2,1}^{(p)}$ | ADMM | [BPC+11] |
| Non-convex | $\min_x (1/2)\|Gx-b\|_2^2 + \lambda\|\mathcal{P}x\|_{p,q}^{(pK)}$ | nmAPG | [LL15] |
|  | $\min_x (1/2)\|Gx-b\|_2^2 + \lambda_1\|\mathcal{P}x\|_{p,q}^{(p)} + \lambda_2\|\mathcal{P}x\|_{p,q}^{(pK)}$ | ADMM | ✗ |
|  | $\min_x (1/2)\|Gx-b\|_2^2 + \lambda_1\|\mathcal{P}x\|_{p,q}^{(p)} + \lambda_2\|\mathcal{D}x\|_{p,q}^{(p)}$ | ADMM | ✗ |

Table 1: The existing global convergence proof for each algorithm to solve each of our formulations.

proximal algorithms depended on the exactness of the proximal operator. Numerical errors on proximal operator computation or the inexact proximal operator may cause the algorithm to diverge. In [GWHH18], [YKG+17], they proposed a way to control the inexactness of computed proximal operator up to some tolerance degree to make the nmAPG algorithm converged. This allows a cheaper computation of the proximal operator. This should be an alternative solution to the ADMM algorithm in our non-convex cases. However, the numerical solution for the proximal operator of the non-convex function may have multiple local optima and some of $p, q$ already have a closed-form expression of proximal operator. For this reason, the splitting technique such as ADMM is preferred over the inexact proximal algorithms. The ADMM algorithm in a non-convex setting is known that its main factor in the convergence issue is its penalty parameters $\rho$. The penalty needed to be large enough [WYZ19],[GHW17] to make the iterations converged. Furthermore, the optimality condition can still be efficiently checked through the primal, dual residuals [BPC+11].

To solve the convergence issue, one obvious way to do is to restart the algorithm with a larger $\rho$. However, a larger $\rho$ may make the optimization progress slower. To see this issue, we replace the ADMM format (40) to

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) + (\rho/2)\|Ax + Bz - c\|_2^2 \\ \text{subject to} & Ax + Bz = c \end{array}$$

which does not change the problem at all. This is the trade-off between the $f(x) + g(z)$, the true objective function, and the primal residuals. If $\rho$ is too small, then the algorithm may generate infeasible sequences, which are regarded as divergence. If $\rho$ is too large, then the minimization focuses heavily on minimizing the primal residuals but not on minimizing its actual objective. This is one of the interpretations of why the convergence is slow when $\rho$ is large. A scheme of adaptive $\rho$ adjustment was introduced to cope with the slow convergence issue. The ADMM with the adaptive regime is proposed in [XFG17], which is called spectral ADMM, but it was applied to solve the convex problems. In non-convex settings, [XDF+16] gives the performance of spectral ADMM over the non-convex problem, and the results yielded that the spectral ADMM also performed well in the non-convex setting.

To conclude this section, we refer to Table 1. The ADMM algorithm has a global convergence in all convex formulations; however, it has no global convergence guaranteed in all of the non-convex formulations. Only non-convex formulation C can be solved using nmAPG,mAPG, PG algorithms. These algorithms have global convergence. Although the ADMM algorithm does not have global convergence, with a proper choice of the algorithm parameter, it can be controlled to have a convergence in practice. For instance, the spectral ADMM may be used to solve these formulations. At this point, the algorithms to solve formulation in the table 1 have been established. In the next section, we implemented these algorithms to investigate the performance of our proposed formulations in the simulated data intensively.

# 6 Preliminary results

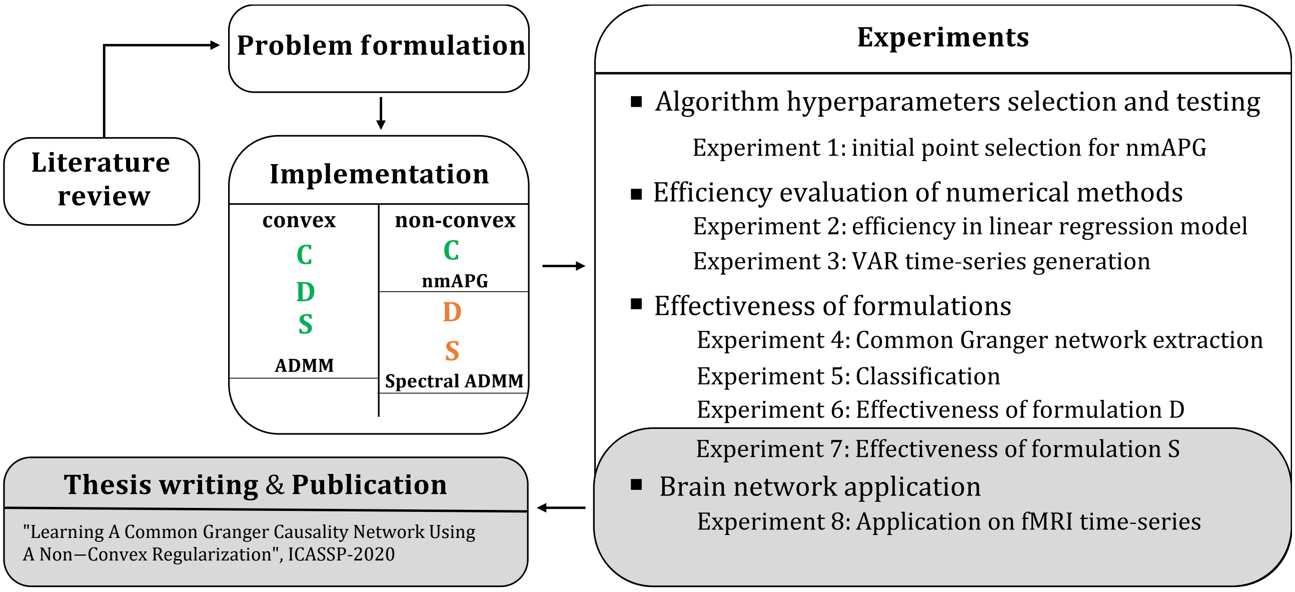
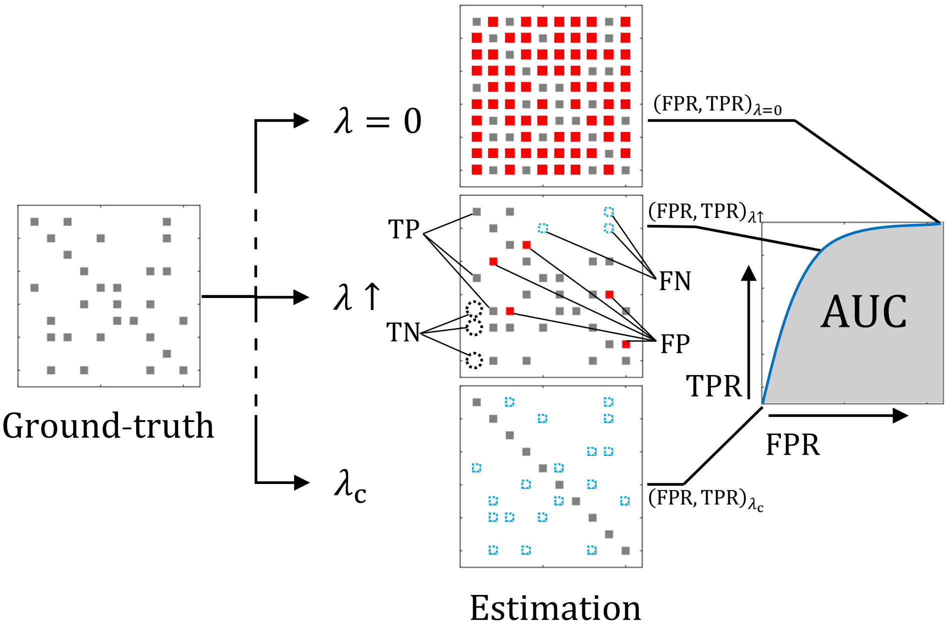

Problem formulation
Literature review
Implementation
convex
C
D
S
ADMM
non-convex
C
nmAPG
D
S
Spectral ADMM
Experiments
Algorithm hyperparameters selection and testing
Experiment 1: initial point selection for nmAPG
Efficiency evaluation of numerical methods
Experiment 2: efficiency in linear regression model
Experiment 3: VAR time-series generation
Effectiveness of formulations
Experiment 4: Common Granger network extraction
Experiment 5: Classification
Experiment 6: Effectiveness of formulation D
Experiment 7: Effectiveness of formulation S
Brain network application
Experiment 8: Application on fMRI time-series
Thesis writing & Publication
"Learning A Common Granger Causality Network Using A Non−Convex Regularization", ICASSP-2020

Figure 9: Progress of work plan in this thesis. The highlighted parts denotes our unfinished works. The formulation highlighted in green in the implementation block can be solved efficiently in a large scale. The algorithm is stated below the formulation. The formulation highlighted in orange is able to solve with spectral ADMM but not as efficient as other cases.

We intended to use this section to illustrate the performance of both formulation and algorithm. The experiments were designed from a work plan stated in Figure 2. We illustrated the progress of this thesis in Figure 9. The shaded area denotes the unfinished tasks. In the implementation block, the green highlight indicates that we have already implemented an efficient algorithm stated below to solve the formulation in a large scale setting. The orange highlight indicates that we have implemented the algorithm, which is stated below the formulations, that make the iterates converged to a critical point but cannot be used in a large scale setting yet.

The experiments are divided into two main parts. First is due to the nature of the non-convexity of the problem, we set experiment 1 to find a good initial guess for initializing the non-convex optimization. After the initialization problem is concluded, we further investigate the performance of the methods in the second part of the experiments. The experiments in this part are experiments 3, 4, 5, 6. In a sparse GC estimation framework, the regularization must be varied in order to evaluate the performance. As shown in Figure 10, the estimation of GC networks in each regularization level can be interpreted as a binary classification, whether each GC connection exists or not. So, there will be two types of classification errors, the false positives and false negatives, and two types of correct classification, true positives, and true negatives. The positive results denote the existence of GC connection. The negative results denote the



Ground-truth
$\lambda = 0$
$\lambda \uparrow$
$\lambda_c$
TP
TN
FN
FP
$(\mathrm{FPR}, \mathrm{TPR})_{\lambda=0}$
$(\mathrm{FPR}, \mathrm{TPR})_{\lambda\uparrow}$
$(\mathrm{FPR}, \mathrm{TPR})_{\lambda_c}$
AUC
TPR
FPR
Estimation

Figure 10: The ROC curve in sparse GC estimation.

absence of GC connection. These accuracy measures at one instance of regularization can be used to compute false positive rate $(\mathrm{FPR})$ and true positive rate $(\mathrm{TPR})$ as

- $\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$
- $\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$

By varying the regularization, the yielded pairs of $\mathrm{FPR}, \mathrm{TPR}$ form a curve called receiver operating characteristic (ROC) curve, as shown in the right-hand-side part of Figure 10. This curve is a tool to illustrate the performance of a method when the sensitivity of a method is varied. The densest GC network will have a $100\%$ true positive rate but also $100\%$ false positive rate, which will be at the upper-right corner of the ROC curve. The sparsest GC network yielded from (17) will have a $0\%$ false positive rate but also $0\%$ true positive rate, which will be at the lower-left corner of the ROC curve. The area under the ROC curve (AUC) can be used as a performance criterion. To see the estimation bias, we also consider the relative parameter bias,

- relative parameter bias $= \frac{\|\hat{x}-x_{\mathrm{true}}\|_2}{\|x_{\mathrm{true}}\|_2}$

In conclusion, the criteria we choose are FPR, TPR, AUC, and relative parameter bias.

As we stated in the algorithm section, the vanilla ADMM algorithm has a convergence issue when solving formulation D. However, we were able to fine-tune the penalty parameter to have convergence. This allowed us to see the performance of the formulations but for limited samples. This problem will be discussed again at the end of this proposal as our future work. For completeness of this section, the experiments we performed are

1. Experiment 1: Initial point selection for nmAPG algorithm.
2. Experiment 2: Non-convex group norm regularization performance in linear regression model.

3. Experiment 3: VAR time-series with pre-specified GC patterns generation
4. Experiment 4: Common Granger network extraction
5. Experiment 5: Supervised-classification using learned common Granger network
6. Experiment 6: Effectiveness of differential prior

In each experiment, the importance of the experiment is discussed in the beginning.

## 6.1 Experiment: Initial point selection for nmAPG algorithm.

**Objective** The non-convex problems are known to have multiple local-minima and heavily affected by the choices of initialization. In these experiments, we aim to select an initial point for solving the non-convex optimization problem (16) when applied to the simple linear regression problem. One of the heuristic approaches was initializing the non-convex problem with the solution of group lasso regression, but we aim to find if there are other easier choices than solving an optimization problem.

**Setting** In this experiment, we investigated the initialization in non-convex formulation (16) in a simple linear regression model. The ground-truth model is defined as

$$b = G\tilde{x} + \epsilon$$

where $b \in \mathbf{R}^{200}, \tilde{x} \in \mathbf{R}^{1000}$ with SNR of $20\text{dB}$. We also assumed that $\tilde{x}$ has structural sparsity, each group of size $10$ and there are $10$ non-zero groups out of 100 groups.

We considered 7 initialization in comparison which are,

- $x_{\text{zero}} = \mathbf{0}$, (zero initialization)
- $x_{\text{ridge}} = (G^TG + 0.1I_{1000})^{-1}G^Tb$, (ridge solution initialization, $\lambda = 0.1$)
- $x_{\text{minnorm}} = G(GG^T)^{-1}b$, (minimum-norm solution initialization)
- $x_{\text{rand}} \sim \mathcal{N}(0, I_{1000})$, (Gaussian iid. randomized zero mean initialization)
- $x_{\text{rand+ridge}} \sim \mathcal{N}(x_{\text{ridge}}, I_{1000})$, (Gaussian iid. randomized with ridge as mean vector initialization)
- $x_{\text{convex}} = \{x | \underset{x}{\text{argmin}} \frac{1}{2}\|Gx - b\|_2^2 + \lambda \|x\|_{2,1}^{(10)}\}$, (convex)
- $x_{\text{Groundtruth}} = \tilde{x}$, (Ground-truth initialization)

We noted that the ground-truth initialization cannot be achieved in practice, we used this as a benchmark to compare with other initialization as the ground-truth initialization should give the best performance.

| Initialization | Loss | Parameters bias |
|---|---|---|
| zero | 5.2445 | 0.5208 |
| ridge | 5.2442 | 0.5208 |
| min-norm | 5.2445 | 0.5208 |
| rand | 5.7040 | 0.7892 |
| ridge+rand | 5.7069 | 0.7931 |
| convex | 5.2183 | 0.4764 |
| Ground truth | **5.2115** | **0.4118** |

Table 2: Average value of loss and relative parameter bias in each initialization over 1000 realization.

**Results** The performance of each initialization can be directly measured by the value of the loss function in (16). Moreover, we investigated the relative model's parameters bias between the learned patterns to confirm whether the lower loss should imply lower error between the ground truth model's parameters and the estimated parameters. To conclude, we selected the performance indicators as

1. Loss function, $(1/2)\|G\hat{x} - b\|_2^2 + \lambda \|\hat{x}\|_{2,1/2}^{(10)}$
2. Relative parameter bias, $\frac{\|\hat{x}-x_{\text{true}}\|_2}{\|x_{\text{true}}\|_2}$.

The initialization that gives the lowest parameter bias will be used in the performance comparison experiments. The result is reported in Table 2. In the table, the ground truth initialization gave the best performance on both loss and relative bias, which support what we hypothesized. The second-best performance is the convex initialization. The results suggested us to select the convex initialization due to ground truth initialization is impractical. As we can see that the initialization from the convex solution gives the second rank of performance, both loss and parameter bias. In the rest experiments, we initialized the algorithm to solve the non-convex formulation with the solution of their convex formulation counterpart.

## 6.2 Experiment: Non-convex group norm regularization performance in linear regression model

**Objective** In this experiment, we explored the performance of the non-convex group norm penalty or the $\ell_{p,q}$ group-norm penalty against the group lasso or the $\ell_{2,1}$ group-norm penalty. The objective is to find out whether the non-convex regularizer outperformed the group lasso and further compare it with the non-group case, which is lasso and $\ell_q$ penalty. We expected that the structural prior of both group lasso and $\ell_{p,q}$ group norm penalty to outperform its non-group counterpart.

**Setting** We generated a model in the same way with the previous experiment or the initialization selection experiment with $\text{SNR} = 20\text{dB}$. We varied the regularization parameter in (24) to yield the densest model to the sparsest model. The sparsest model is yielded from the minimum regularization that gives all zero solutions in the group lasso case. The non-convex formulation also varied in the same range because the non-convex formulation tends to be sparser. So that, the regularization bound of the convex case is also a sufficient condition in the non-convex case. The range in $\ell_1, \ell_q$ is set in the same sense. We repeated this experiment 10 times. The area under the ROC curve is the indication we selected to measure the performance between these two regression methods.

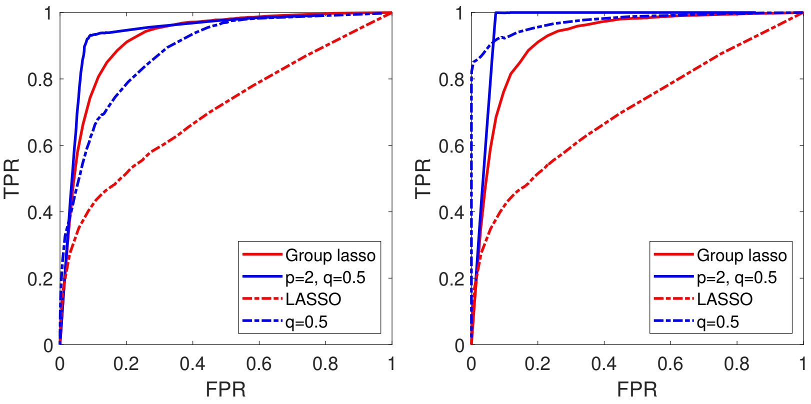


Figure 11: ROC curve of linear regression using $\ell_{1/2}$, $\ell_{2,1/2}$ regression, lasso and group lasso. nmAPG was initialized by solution of (left) $\ell_{2,1}$ regression, (right) ground truth parameters.

**Results** The result in the figure 11 suggested that the regularized regression with $\ell_{2,1/2}$ outperformed $\ell_{2,1}$, which agreed with the previous result in the literature. Moreover, both group and non-group non-convex penalties have significant improvement when using the ground-truth model as an initialization as shown in the right plot of Figure 11. This result indicated that there would be a room for improvement for initial point selection. The effectiveness of group penalties are evidently shown in the left plot of Figure 11. Even $\ell_q$ was outperformed by the group lasso. So, if the good initialization cannot be found, the penalty with a grouping structure should be considered first.

### 6.3 Experiment: VAR time-series with pre-specified GC patterns generation

**Objectives** In this experiment, we aim to generate multiple VAR models with pre-specified GC patterns in both common parts and differential parts. The generated models will be used to generate time-series for other experiments.

**Setting** To generate a stable VAR model with dimension $(n, p, K)$ where $n$ is the time-series dimension, $p$ is lag number, $K$ is the number of models, we first define the stability of VAR processes. In single VAR model of order $p$, The generated VAR processes is stable if and only if the matrix,

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix},$$

has eigenvalue inside unit circle. We exploited the special case when each $A_r$ is a diagonal matrix, so that the characteristic equation of this matrix is

$$\prod_{i=1}^{n} (z^p (A_1)_{(i,i)} + z^{p-1} (A_2)_{(i,i)} + \cdots + (A_p)_{(i,i)}) = 0$$

We randomized the diagonal matrix $A_r$ such that the roots of this characteristic equation stay inside unit circle to guarantee stability of VAR processes. We repeated this $K$ times to yield $K$ VAR models. We refer this as the diagonal VAR model. However, we also assume that the $K$ VAR models have three types of relation to mimic our assumption on the formulation. The assumptions are

1. Common type ground truth: All $K$ models have Identical pattern of GC network. The value of coefficients can be different,
2. Differential type ground truth: All $K$ models partially share pattern of GC network but each model also has its own different pattern. The density of shared pattern is referred as **common density** and the density of different pattern is referred as **differential density**,
3. Similar type ground truth: this assumption is the same as the differential ground truth but the shared pattern also have the same VAR coefficients.

After we yielded the diagonal VAR models, we randomized off-diagonal coefficients of $K$ to have GC networks pattern as described which are based on the condition (4).

In our settings, the models parameters are

- $n = 15, p = 2, K = 4$,
- common density = 0.1, 0.2,
- differential density = 0.01, 0.05,
- spectral radius $\sim \mathcal{U}(-0.7, 0.7)$

The time-series generation is simple. We generated multivariate time-series of the generated model for $4500$ time points with iid. unit variance Gaussian noise.

**Result** In this experiment, we discuss how we randomly generated stable VAR processes. However, this method is not efficient due to its stability criterion. If there exists a dense connection in the GC pattern or the model is in high order, the stability of the model is hard to achieve. To the best of our knowledge, the dense model cannot be easily constructed. However, dense models are not of our interests.

### 6.4 Experiment: Group-level Granger network extraction

**Objectives** In this experiment, we investigate the performance of formulation C (16) under the simulated time-series data with given Granger networks that decomposed into common and differential networks. We intended to use this formulation to extract the group-level Granger causality of a panel time-series data. In real data applications, the panel data can be time-series of each brain region for multiple patients. The extraction of a common network can be applied to fetch the group-level brain connectivity of patients with brain disease.

**Setting** We generated GC networks with a common density of 0.1 and 0.2 and the differential density of 0.01 and 0.05 by the method described in the data generation section. We set $n = 15, p = 2, K = 4$ in this experiment. To see the performance, we also varied the time-series length, which is the sample size to be $T = 50, 300, 1350$. The sparsity level is varied from the densest solution to the sparsest solution. The regularization level that gives the sparsest solution is derived in [Son17].

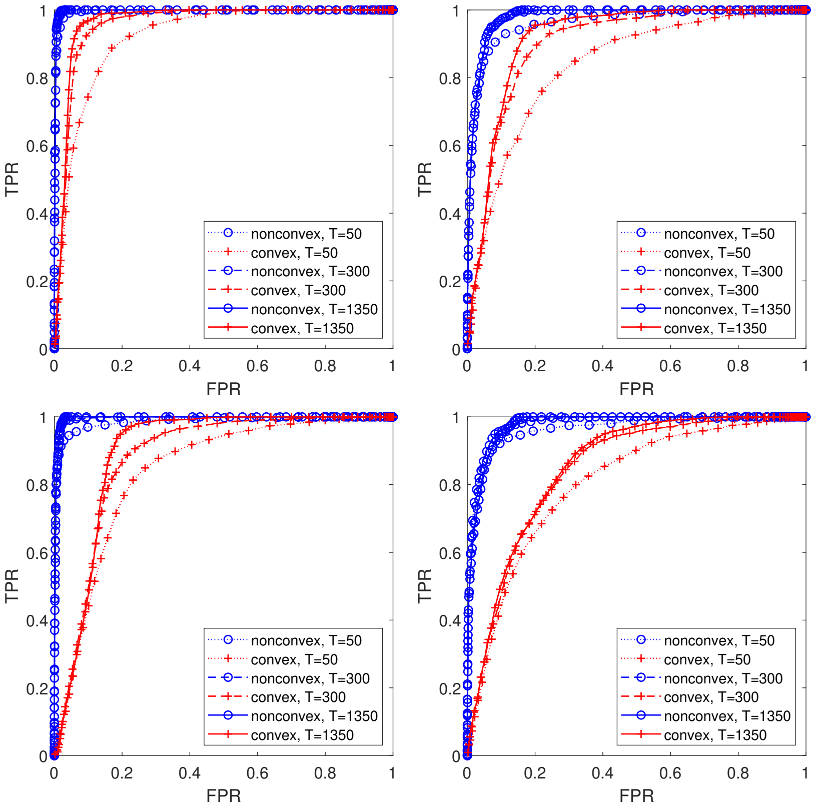


Figure 12: The ROC curve comparison between **group lasso** and $\boldsymbol{\ell_{2,1/2}}$ penalty. Common density is varied as **0.1, 0.2** (upper-lower). Differential density is varied as **0.01, 0.05** (left-right).

**Results** The figure 6.4 shown ROC curves of the joint estimation formulations based on both $\ell_{2,1/2}$ and group lasso penalty. We used only a common GC network part of the ground-truth model to evaluate accuracy. The results suggested that the networks learned from $\ell_{2,1/2}$ regularized estimation have a larger area under the ROC curve than the networks that were extracted using group lasso regularized estimation. We also noticed that even with a smaller sample size, the non-convex formulation is still outperformed the convex one. The performance of both convex and non-convex penalty is dropped when the model density is increased. This is a result of a poor choice of prior information about the sparsity of the true model. From the ROC curves, the results yielded from convex formulation were heavily sensitive to the increment of model density, while the non-convex formulation was more robust to the increment in model density.

## 6.5 Experiment: Supervised-classification using learned common Granger network

**Objective** In the previous experiment, we showed that the non-convex regularization improved the recovery of a sparse common Granger network of multiple models. Since a common Granger network can be represented as a Granger network of a class of data. The learned common Granger network can be used to represent those classes if there are multiple classes of data. So the supervised-classification scheme based on a likelihood ratio test is possible.

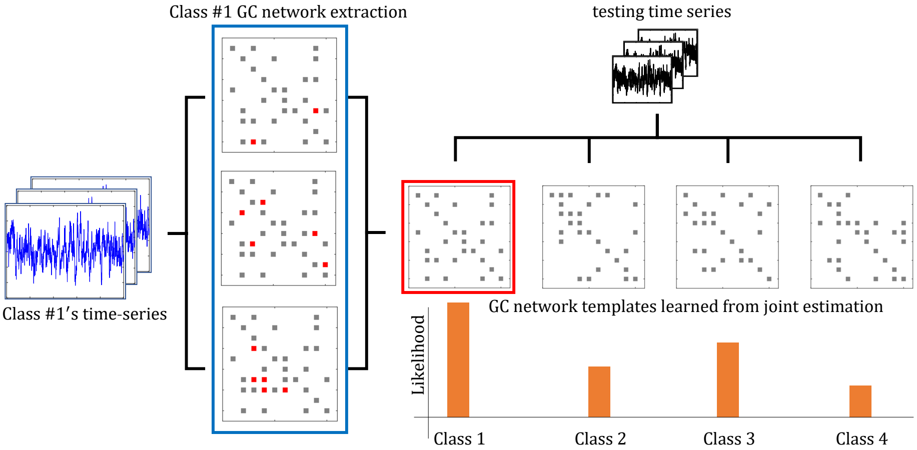

Class #1 GC network extraction
testing time series
Class #1′s time-series
GC network templates learned from joint estimation
Likelihood
Class 1
Class 2
Class 3
Class 4

Figure 13: Supervised classification scheme.

**Setting** The overall process is given in Figure 13. The idea behind this scheme is intuitively simple. First, we suppose that there are $M$ classes of data; each class has $K$ sets of $n$-dimensional time-series that shared a common Granger network. The common Granger network is extracted by our formulation and group lasso and used as a template for each class. The unknown time-series is fitted into a multi-classes VAR model with the sparsity pattern constraint of each class. The template that explains the unknown time-series the most in the sense of highest likelihood, then the unknown time-series belongs to that class.

In this experiment, we generated $10$ classes of multiple $15$-dimensional second-order VAR processes with given Granger networks. The common network's density is set to be $20\%$. We followed the described classification routine. However, we also varied the model order to see whether the wrongly chosen model order significantly degrades the performance or not.

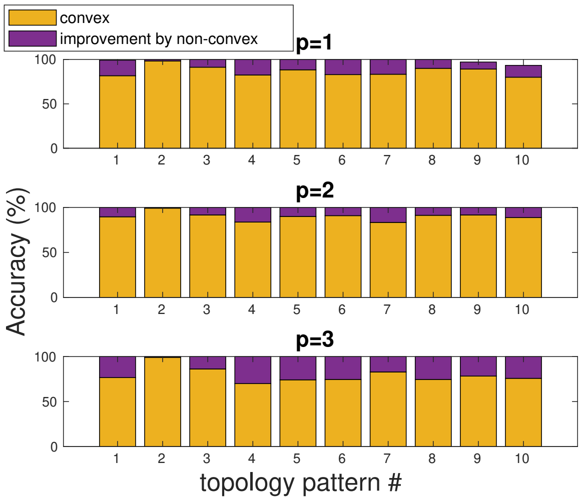


Figure 14: Classification accuracy

**Results** In Figure 14, the classification using GC template learned with $\ell_{2,1/2}$ regularization has significant accuracy improvement from the classification in the group lasso case. With the $\ell_{2,1/2}$ penalty, the classification has a near-perfect classification rate. Moreover, even if the model order $p$ is wrongly chosen, the performance does not decay. The result in this experiment suggested that the template extracted by $\ell_{2,1/2}$ regularized joint estimation outperformed the template in the group lasso case.

### 6.6 Experiment: Effectiveness of differential prior

**Objective** In this experiment, we investigated the performance of our non-convex formulation or formulation D (19) to our generated data sets. Since the convergence of the ADMM algorithm is unknown, we fine-tuned the penalty parameter and monitoring the primal residuals and dual residuals. We applied our penalty in (19), and the convex penalty (18) to see whether the performance of the non-convex group norm penalty outperforms its convex counterpart or not. We varied $\lambda_1, \lambda_2$ from densest solution to the sparsest solution. The tuning parameter $\lambda_1$ controls the differential sparsity pattern, while $\lambda_2$ controls the common sparsity pattern.

**Setting** In this experiments, we generated three types of ground-truth models as a result from Experiment 3. The ground-truth types are

1. Common type ground-truth
2. Differential type ground-truth
3. Similar type ground-truth

In all types, we generated four sets of $15$-dimensional VAR models. In each type, the ground-truth GC networks were designed to have patterns the same as in the formulations. In the common type ground-truth, all models have identical GC networks. In the differential type ground-truth, the GC networks have both common and differential network. In the similar type ground-truth, all models have a common part with identical coefficient and some differential networks. The vanilla ADMM algorithm is used to solve this problem with fine-tuned algorithm parameter to have a convergence to critical point.

**Results** The convex formulation results are shown in Figure 15 and non-convex formulation in Figure 16. From both figures, the subplots in the left column are ROC curves yielded from comparing a common GC network extracted from the estimation to the ground-truth model's common GC network. The middle subplot is an ROC curve yielded from comparing an estimated differential GC network to the ground-truth model's differential GC network. The ROC curves in the right column yielded from a direct comparison between the ground-truth GC network and the estimated GC network. In the common type ground-truth, the differential network's ROC is null because of the common type ground-truth model does not have differential parts to compare. In each subplot, the ROC curve is constructed by fixing $\lambda_2$ and varying $\lambda_1$. The darker red denotes the smaller value of $\lambda_2$. The trend between AUC and $\lambda_2$ is apparent in Figure 15. The subplot (2,3) in Figure 15 shows that when $\lambda_2$ increases to a suitable value that results in an appropriate level of sparsity in the common GC network, we will see the ROC with the highest AUC. However, as we keep increasing $\lambda_2$, this does not necessarily improve AUC because if $\lambda_2$ is too large, the common GC network is entirely sparse, causing all entries in VAR coefficients zero. In that case, the effect of $\lambda_1$ for promoting differential sparsity in each model becomes trivial, as we can see in the brighter-tone ROC having a small AUC. The ROC of the left and middle column may be a loop because we evaluated the common sparsity and differential sparsity independently, but the common and differential networks are related when varying the

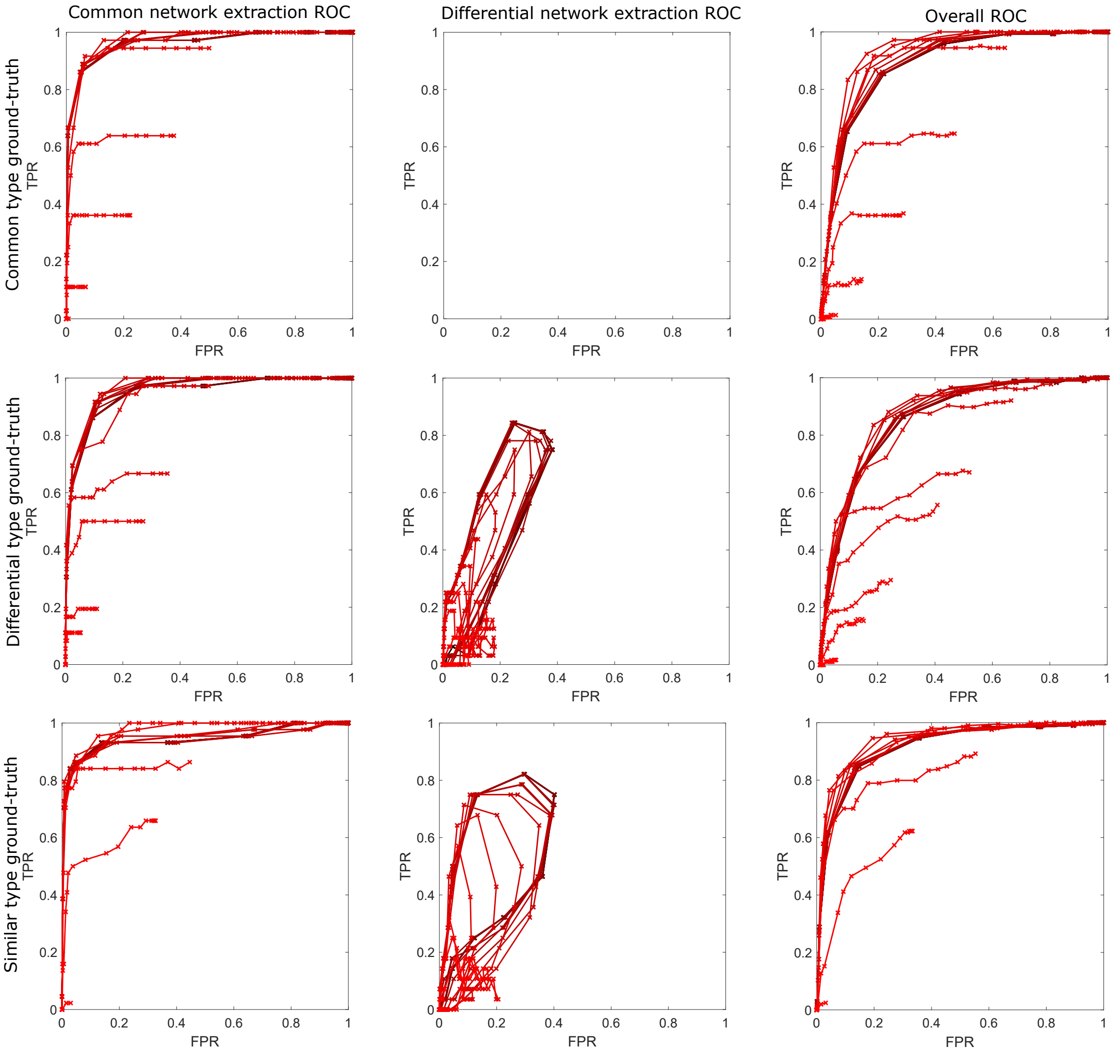


Figure 15: The ROC curve of learned GC networks using **convex penalty**.

regularization. Normally, the ROC curve starts at $(\mathrm{FPR}, \mathrm{TPR}) = (0, 0)$ or the diagonal GC matrix case and ends at $(1, 1)$ or the densest GC matrix case. If there is a GC connection for every pair of variables, then every connection is a common network without any differential GC networks because all connections exist in common in all models. So, the differential ROC starts at $(0, 0)$. by increasing $\lambda_1$, the differential networks are encouraged. However, if $\lambda_1$ is too large, then all differential networks are removed, so the differential ROC ends at $(0, 0)$. This is the reason why the ROC curve is a loop in differential ROC curve in Figure 15 and Figure 16. The reason behind the difference between the common network ROC and the overall ROC in a common type ground-truth is that the number of true negatives is not equal, while the number of true positives is equal. By direct comparison of AUC, the results show that the non-convex cases outperformed the convex case at all ground-truth types in the sense of having more area under the ROC curve.

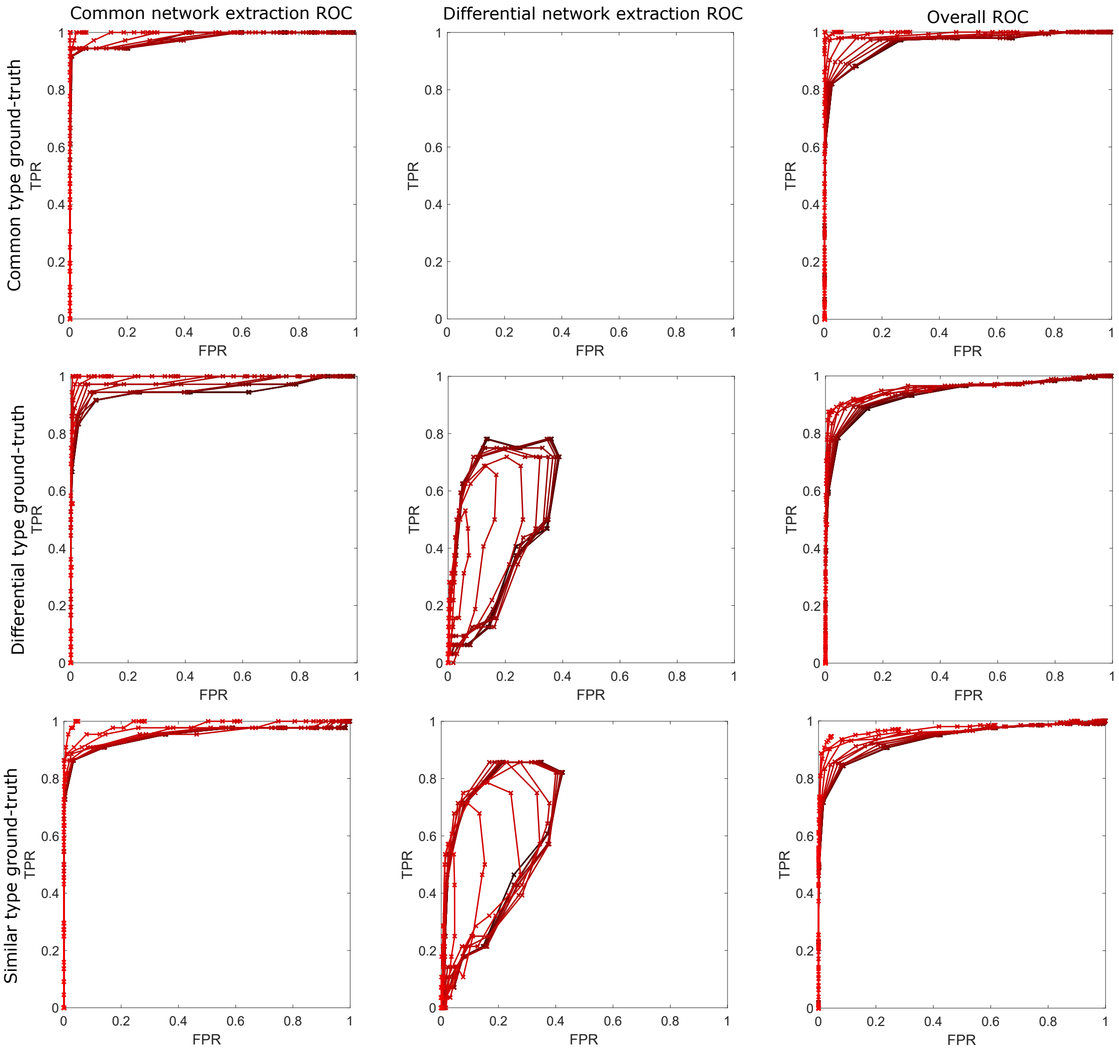


Figure 16: The ROC curve of learned GC networks using **non-convex penalty**.

# 7 Future works

We proposed joint estimation formulations to estimate multiple Granger graphical models with non-convex block-sparse inducing penalties. The networks decomposed to homogeneity and heterogeneity parts. As shown in Figure 9, we performed the experiments that are not shaded. Those experiments are carefully designed to verify the usefulness of the proposed formulation step by step. In the first experiment, we answered our research question on how we should guess the initialization of our non-convex optimization problem. A good guess we found is the solution of the convex optimization problem counterpart. After that, the performance of non-convex group sparse regularization is verified in our second experiment. The experiment 4, 5, and finally 6, are the heart of this thesis. These experiments were set up to provide clarification to us for answering the question, *will our formulation work?* All numerical results suggested that our formulation performed better than the convex formulations proposed in [Son17] in the sense of having more area under the ROC curve. However, this question is only partially answered. Experiment 6 cannot be performed intensively and, hence, cannot be used in a large scale setting yet. The available and efficient algorithm for this problem is required to answer the question. Both algorithm performance and coding efficiency are crucial to our work not only in theoretical but also in a practical aspect. In real data applications such as fMRI data analysis of brain connectivity, there will be a large number of brain regions to identify their causal relations. As stated in our work plan, we will further optimize the algorithm and develop a program to perform experiments in a large scale setting. Moreover, from the partial success in our simulation data sets, we aim to extract group-level brain connectivity from fMRI data as our future work.

A part of this proposal's preliminary results based on formulation C (24) was presented at ICASSP conference [MS20]. The published work is based on experiment 3, experiment 4, and experiment 5.

# 8 References

[ABD13] C. M. Alaíz, A. Barbero, and J. R. Dorronsoro. Group fused lasso. In *Artificial Neural Networks and Machine Learning – ICANN 2013*, pages 66–73. Springer Berlin Heidelberg, 2013.

[ABS13] H. Attouch, J. Bolte, and B.F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137:91–129, 2013.

[BBS09] L. Barnett, A. B. Barnett, and A. K. Seth. Granger causality and transfer entropy are equivalent for Gaussian variables. *Physical Review Letters*, 103:238701, Dec 2009.

[BLH$^+$20] J. C. Bore, P. Li, D. J. Harmah, F. Li, D. Yao, and P. Xu. Directed EEG neural network analysis by LAPPS (p$\leq$1) penalized sparse Granger approach. *Neural Networks*, 124:213 – 222, 2020.

[BPC$^+$11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *foundation and trends in machine learning*, 3(1):1–122, January 2011.

[BS14] L. Barnett and A. K. Seth. The MVGC multivariate Granger causality toolbox: A new approach to Granger-causal inference. *Journal of Neuroscience Methods*, 223:50 – 68, 2014.

[BS15] L. Barnett and A. K. Seth. Granger causality for state-space models. *Physical Review E*, 91:040101, Apr 2015.

[BT09] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2:183–202, 01 2009.

[CHY16] C. Chen, B. He, and Y. Ye. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155:57–79, 2016.

[CP11] P. L. Combettes and J. C. Pesquet. *Proximal Splitting Methods in Signal Processing*, pages 185–212. Springer New York, New York, NY, 2011.

[CZZ15] H. Chun, X. Zhang, and H. Zhao. Gene regulation network inference with joint sparse Gaussian graphical models. *Journal of Computational and Graphical Statistics*, 24(4):954–974, 2015.

[DF03] D.S. Dummit and R.M. Foote. *Abstract Algebra*. Wiley, 2003.

[Fri11] K. J. Friston. Functional and effective connectivity: A review. *Brain Connectivity*, 1(1):13–36, 2011.

[GHW17] K. Guo, D. R. Han, and T. T. Wu. Convergence of alternating direction method for minimizing sum of two nonconvex functions with linear constraints. *International Journal of Computer Mathematics*, 94(8):1653–1669, 2017.

[GKMM15] M. Gregorova, A. Kalousis, and S. Marchand-Maillet. Learning coherent Granger-causality in panel vector autoregressive models. In *Proceedings of the Demand Forecasting Workshop of the 32nd International Conference on Machine Learning*. ICML, 2015.

[GLMZ11] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 02 2011.

[GWHH18] B. Gu, D. Wang, Z. Huo, and H. Huang. Inexact proximal gradient methods for non-convex and non-smooth optimization. In *AAAI Conference on Artificial Intelligence*, 2018.

[Hau12] S. Haufe. *Towards EEG source connectivity analysis*. Doctoral Thesis, Technische Universität Berlin, Fakultät IV - Elektrotechnik und Informatik, Berlin, 2012.

[HLM$^+$17] Y. Hu, C. Li, K. Meng, J. Qin, and X. Yang. Group sparse optimization via $\ell_{p,q}$ regularization. *Journal of Machine Learning Research*, 18(30):1–52, 2017.

[HLR16] M. Hong, Z. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[HMXZ09] J. Huang, S. Ma, H. Xie, and C. Zhang. A group bridge approach for variable selection. *Biometrika*, 96(2):339–355, 2009.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[LL15] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 379–387. Curran Associates, Inc., 2015.

[LP15] G. Li and T. K. Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.

[Lüt05] H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 2005.

[MB10] N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[MS20] P. Manomaisaowapak and J. Songsiri. Learning a common Granger causality network using a non-convex regularization. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1160–1164, 2020.

[MSMC15] T. Möllenhoff., E. Strekalovskiy., M. Moeller., and D. Cremers. The primal-dual hybrid gradient method for semiconvex splittings. *SIAM Journal on Imaging Sciences*, 8(2):827–857, 2015.

[OCBP14] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[PB14] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, January 2014.

[SM19a] A. Skripnikov and G. Michailidis. Joint estimation of multiple network Granger causal models. *Econometrics and Statistics*, 10:120–133, 2019.

[SM19b] A. Skripnikov and G. Michailidis. Regularized joint estimation of related vector autoregressive models. *Computational Statistics & Data Analysis*, 139:164–177, 2019.

[Son13] J. Songsiri. Sparse autoregressive model estimation for learning Granger causality in time series. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3198–3202, 2013.

[Son17] J. Songsiri. Estimations in Learning Granger Graphical Models with Application to fMRI Time Series. Technical report, Chulalongkorn University, Department of Electrical engineering, July 2017.

[ST19] S. Sabach and M. Teboulle. Chapter 10 - Lagrangian methods for composite optimization. In *Processing, Analyzing and Learning of Images, Shapes, and Forms: Part 2*, volume 20 of *Handbook of Numerical Analysis*, pages 401 – 436. Elsevier, 2019.

[TB20] J. Tuck and S. Boyd. Fitting laplacian regularized stratified gaussian models. *ArXiv*, abs/2005.01752, 2020.

[TY10] K. C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6, 09 2010.

[WB14] H. Wang and A. Banerjee. Bregman alternating direction method of multipliers. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2816–2824. Curran Associates, Inc., 2014.

[WBC18] I. Wilms, L. Barbaglia, and C. Croux. Multiclass vector auto-regressive models for multistore sales data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(2):435–452, 2018.

[WCLQ18] F. Wen, L. Chu, P. Liu, and R. C. Qiu. A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning. *IEEE Access*, 6:69883–69906, 2018.

[WCX18] F. Wang, W. Cao, and Z. Xu. Convergence of multi-block Bregman ADMM for nonconvex composite problems. *Science China Information Sciences*, 61(12):122101:1–122101:12, 2018.

[WYZ19] Y. Wang, W. Yin, and J. Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, Jan 2019.

[XCXZ12] Z. Xu, X. Chang, F. Xu, and H. Zhang. $l_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on neural networks and learning systems*, 23(7):1013–1027, 2012.

[XDF$^{+}$16] Z. Xu, S. De, M. Figueiredo, C. Studer, and T. Goldstein. An empirical study of ADMM for nonconvex problems. In *Workshop on Nonconvex Optimization for Machine Learning: Theory and Practice*, 12 2016.

[XFG17] Z. Xu, M. Figueiredo, and T. Goldstein. Adaptive ADMM with spectral penalty parameter selection. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 718–727, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[YKG$^{+}$17] Q. Yao, J. T. Kwok, F. Gao, W. Chen, and T. Liu. Efficient inexact proximal gradient algorithm for nonconvex problems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3308–3314, 2017.

[ZMZ$^{+}$13] W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang. A generalized iterated shrinkage algorithm for non-convex sparse coding. In *2013 IEEE International Conference on Computer Vision*, pages 217–224, 2013.

[ZQG16] S. Zhang, H. Qian, and X. Gong. An alternating proximal splitting method with global convergence for nonconvex structured sparsity optimization. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2330–2336, 2016.

[ZS19] T. Zhang and Z. Shen. A fundamental proof of convergence of alternating direction method of multipliers for weakly convex optimization. *Journal of Inequalities and Applications*, 2019(128), 2019.

# Appendix

## A Definition of $\mathcal{P}, \mathcal{D}$

In our formulations, the transformation $\mathcal{P}$ is a projection mapping that extracts all off-diagonal entries of VAR coefficients. The transformation $\mathcal{D}$ is a difference matrix transformation that maps VAR coefficients of each model into the difference of off-diagonal VAR coefficients between each model. This is because of the time-series should be dependent on itself so the diagonal entries of VAR coefficients should not be regularized. For example, we consider a case when $x = (C_{11}, C_{12}, C_{21}, C_{22})$ with $C_{ij} \in \mathbf{R}^{pK}$. A projection matrix $\mathcal{P}$ is defined by

$$\mathcal{P} = \begin{bmatrix} 0 & I_{pK} & 0 & 0 \\ 0 & 0 & I_{pK} & 0 \end{bmatrix} \tag{50}$$

so that $\mathcal{P}x = (C_{12}, C_{21})$ which is the off-diagonal entries in GC matrix. The matrix $\mathcal{P}$ has dimension $\mathbf{R}^{(n^2-n)pK \times n^2pK}$ since the diagonal entries of all $p$ lag VAR coefficient matrix of size $n \times n$ in all $K$ models are removed from the projected space. The generalization of this case to $n$-dimensional case is evident.

From formulation S (21), the regularization can be thought of a difference matrix multiply with each $C_{ij}$. The following example is the simple case when $K = 3$. We express $C_{ij}$ in the form of (14) to make it explicit.

$$\begin{bmatrix} B_{ij}^{(1)} - B_{ij}^{(2)} \\ B_{ij}^{(1)} - B_{ij}^{(3)} \\ B_{ij}^{(2)} - B_{ij}^{(3)} \end{bmatrix} = \begin{bmatrix} I_p & -I_p & 0 \\ I_p & 0 & -I_p \\ 0 & I_p & -I_p \end{bmatrix} \begin{bmatrix} B_{ij}^{(1)} \\ B_{ij}^{(2)} \\ B_{ij}^{(3)} \end{bmatrix} := \mathcal{D}_{ij} C_{ij}$$

For the same reason in the projection case, we interest only the case when $i \neq j$, the difference matrix is defined as

$$\tilde{\mathcal{D}} = \operatorname{diag}(\mathcal{D}_{12}, \mathcal{D}_{13}, \ldots, \mathcal{D}_{n-2\ n}, \mathcal{D}_{n-1\ n}) \tag{51}$$

where $\tilde{\mathcal{D}} \in \mathbf{R}^{(n^2-n)p\binom{K}{2} \times (n^2-n)pK}$. If prior information on which GC connections are indeed significant or insignificant is available, the identity matrix in both $\mathcal{D}_{ij}, \mathcal{P}$ can be replaced with arbitrary positive scaling of identity matrix of the same size. This allows adaptive regularization framework for the estimation. When all $\mathcal{D}_{ij}$ are identical to $\mathcal{D}_c$, the matrix $\tilde{\mathcal{D}}$ is

$$\tilde{\mathcal{D}} = I_n \otimes \mathcal{D}_c$$

where $\otimes$ is a kronecker product. For a more compact representation, we define

$$\mathcal{D} = \tilde{\mathcal{D}}\mathcal{P}$$

so that $\mathcal{D}x$ is the group differences of the off-diagonal coefficients only.