

# KOO Method–based Consistent Clustering for Group-wise Linear Regression with Graph Structure

Mineaki Ohishi<sup>1\*</sup> and Ryoya Oda<sup>2</sup>

<sup>1</sup>Center for Data-driven Science and Artificial Intelligence, Tohoku University  
Kawauchi 41, Aoba-ku, Sendai 980-8576, Japan

<sup>2</sup>Graduate School of Advanced Science and Engineering, Hiroshima University  
1-3-1 Kagamiyama, Higashi-Hiroshima 739-8526, Japan

## Abstract

The kick-one-out (KOO) method is a variable selection method based on a model selection criterion. The method is very simple, and yet it has consistency in variable selection under a high-dimensional asymptotic framework with a specific model selection criterion. This paper proposes the join-two-together (JTT) method, which is a clustering method based on the KOO method for group-wise linear regression with graph structure. The JTT method formulates the clustering problem as an edge selection problem for a graph and determines whether to select each edge based on the KOO method. We can employ network Lasso to perform such a clustering. However, network Lasso is somewhat cumbersome because there is no good algorithm for solving the associated optimization problem and the tuning is complicated. Therefore, by deriving a model selection criterion such that the JTT method has consistency in clustering under a high-dimensional asymptotic framework, we propose a simple yet powerful method that outperforms network Lasso.

(Last Modified: September 14, 2025)

**Key words:** Clustering, Consistency, Generalized  $C_p$  criterion, High-dimension, Kick-one-out method, Model selection.

\*Corresponding author

E-mail address: mineaki.ohishi.a4@tohoku.ac.jp (Mineaki Ohishi)

## 1. Introduction

For  $m$  groups, suppose we have a dataset  $(\mathbf{y}_j, \mathbf{X}_j)$  ( $j \in \{1, \dots, m\}$ ), where  $\mathbf{y}_j$  is an  $n_j$ -dimensional vector of a response variable,  $\mathbf{X}_j$  is an  $n_j \times p$  matrix of explanatory variables satisfying  $\text{rank}(\mathbf{X}_j) = p \leq n_j$ , and  $n_j$  is the sample size of the  $j$ th group. For such a dataset, we then assume the following group-wise linear regression model:

$$\mathbf{y}_j \sim N_{n_j}(\mathbf{X}_j\boldsymbol{\beta}_j, \sigma^2\mathbf{I}_{n_j}), \quad (1.1)$$

where  $\beta_j$  is a  $p$ -dimensional vector of regression coefficients,  $\sigma^2$  is an error variance satisfying  $\sigma > 0$ , and  $\mathbf{y}_1, \dots, \mathbf{y}_m$  are mutually independent. Furthermore, we assume  $N - 4 > 0$ , where  $N = n - mp$  and  $n = \sum_{j=1}^m n_j$ . A simple method for estimating  $\beta_1, \dots, \beta_m$  is to apply the ordinary least squares (OLS) or maximum likelihood estimation method group-wise. The group-wise estimator of  $\beta_j$  ( $j \in \{1, \dots, m\}$ ) obtained from either method is as follows:

$$\hat{\beta}_j = \mathbf{M}_j^{-1} \mathbf{X}'_j \mathbf{y}_j, \quad \mathbf{M}_j = \mathbf{X}'_j \mathbf{X}_j. \quad (1.2)$$

If  $m$  groups have relationships with each other, it would be desirable to utilize an estimation method that considers those relationships rather than the simple group-wise estimation method given above. In the present paper, we assume that the relationships among the groups can be formulated as a graph. That is, we assume that the relationships are given by a graph  $(V, E)$  with a vertex set  $V = \{1, \dots, m\}$  and an edge set  $E \subset V \times V$  satisfying  $(k, \ell) \in E \Rightarrow k < \ell$ . For example, by regarding houses or areas as groups, we can define  $E$  in terms of regional adjacency (e.g., Hallac *et al.*, 2015; Ohishi *et al.*, 2025). If  $(k, \ell) \in E$ , the  $k$ th and  $\ell$ th groups are related, and if  $(k, \ell) \notin E$ , they are not related. As a method for estimating  $\beta_1, \dots, \beta_m$  under such a graph structure, we can consider a penalized estimation method with the penalty based on differences of regression coefficient vectors for related groups, i.e.,  $\beta_k - \beta_\ell$  ( $(k, \ell) \in E$ ). Specifically, we can use network Lasso (Hallac *et al.*, 2015) to obtain estimators satisfying  $\beta_k = \beta_\ell$  exactly for some  $(k, \ell) \in E$ . The network Lasso estimator is obtained by minimizing the following penalized residual sum of squares:

$$\sum_{j=1}^m \|\mathbf{y}_j - \mathbf{X}_j \beta_j\|^2 + \lambda \sum_{(k, \ell) \in E} w_{k\ell} \|\beta_k - \beta_\ell\|, \quad (1.3)$$

where  $\lambda$  ( $\geq 0$ ) is a tuning parameter adjusting the strength of the penalty against the model fitting and  $w_{k\ell}$  ( $> 0$ ) is a penalty weight expressing the strength of the relationship between the  $k$ th and  $\ell$ th groups. Network Lasso can perform an estimation that considers the relationships among the groups by shrinking  $\beta_k - \beta_\ell$  based on a network structure. Notably, it allows  $\|\beta_k - \beta_\ell\|$  to be exactly zero. This implies that  $\beta_k$  and  $\beta_\ell$  can be estimated as exactly equal. In this case, we can perform clustering by interpreting that the  $k$ th and  $\ell$ th groups belong to the same cluster. However, network Lasso is somewhat cumbersome in practice. One of its problems is the optimization method. If clustering is not required, then the optimization method is not a problem. For example, we can adopt  $\|\beta_k - \beta_\ell\|^2$  instead of  $\|\beta_k - \beta_\ell\|$  in (1.3). In this case, we can perform an estimation based on a network structure and the estimator can be obtained in closed form. However, since the network Lasso estimator cannot be obtained in closed form, its optimization method is important. Hallac *et al.* (2015) proposed an algorithm based on the alternating direction method of multipliers (ADMM; Boyd *et al.*, 2011) to minimize (1.3). ADMM is a very popular method because it is highly versatile and has good theoretical properties, e.g., optimality and convergence. On the other hand, it too has some problems in practice. We focus here on two problems: the convergence of the algorithm is slow and  $\beta_k$  and  $\beta_\ell$  cannot be estimated as numerically and exactly equal for network Lasso. The latter in particular is serious

because clustering is one purpose of applying network Lasso. In order to overcome these problems in ADMM, Ohishi *et al.* (2025) proposed an algorithm based on the block-wise coordinate descent method (BCDM). BCDM can estimate  $\beta_k$  and  $\beta_\ell$  as numerically and exactly equal and Ohishi *et al.* (2025) reported that BCDM is superior to ADMM in terms of optimization speed and minimization accuracy. However, since the penalty is not separable with respect to  $\beta_j$ , BCDM is not guaranteed to have the desired theoretical properties, e.g., optimality and convergence. As mentioned above, for network Lasso, there is not an algorithm with both theoretical and practical goodness. In addition, since network Lasso requires the selection of  $\lambda$  and  $w_{k\ell}$ , network Lasso cannot be said to be easy to use.

The present paper focuses on clustering based on a graph and considers an approach that is completely different from continuous optimization methods such as network Lasso. Specifically, we focus on the kick-one-out (KOO) method (Zhao *et al.*, 1986; Nishii *et al.*, 1988), which is a variable selection method (the name “kick-one-out” was given by Bai *et al.*, 2018). The KOO method determines whether to select the  $j$ th variable by comparing a model which has all  $p$  variables with a model excluding only the  $j$ th variable and the goodness for each model is evaluated by a model selection criterion, e.g., the Akaike information criterion (Akaike, 1973) or the  $C_p$  criterion (Mallows, 1973). Hence, the KOO method requires only  $p$  calculations and is feasible in a large number of variables. Despite the KOO method being very simple, as just described, it has consistency in variable selection. For example, for the problem of selecting explanatory variables in multivariate linear regression, Oda & Yanagihara (2020, 2021) revealed classes of the generalized  $C_p$  ( $GC_p$ ) criterion (Atkinson, 1980) and generalized information criterion (Nishii, 1984), respectively, such that the KOO method has consistency in variable selection under a high-dimensional asymptotic framework, and proposed specific criteria. Furthermore, Oda & Yanagihara (2020) reported that the KOO method is superior to adaptive group Lasso (Wang & Leng, 2008), which is a continuous optimization method, in terms of performance of variable selection and calculation time. Inspired by the KOO method, we propose the join-two-together (JTT) method, which performs clustering for group-wise linear regression. The JTT method formulates the clustering problem as the problem of selecting the pairs  $(k, \ell) (\in E)$  that should belong to the same cluster. In other words, the clustering problem is formulated as the problem of selecting edges  $(k, \ell)$ , where if  $(k, \ell)$  is selected, the  $k$ th and  $\ell$ th groups are interpreted as belonging to the same cluster. Specifically, the JTT method determines whether to select the edge  $(k, \ell)$  (i.e., whether  $\beta_k = \beta_\ell$ ) by comparing a model in which all regression coefficient vectors are different with a model in which  $\beta_k = \beta_\ell$  only for  $(k, \ell)$ . When  $(V, E)$  is a complete graph,  $\#(E)$  achieves the maximum and is  ${}_m C_2 = m(m-1)/2$ . Hence, the JTT method requires a calculation of  $O(m^2)$ . In the present paper, the  $GC_p$  criterion is employed to evaluate the goodness of a model. Based on Oda & Yanagihara (2020), we reveal a class of the  $GC_p$  criterion such that the JTT method has consistency in the edge selection under a high-dimensional asymptotic framework and propose a specific criterion. Moreover, we show that the JTT method is superior to network Lasso through numerical studies.

The remainder of the paper is organized as follows. In Section 2, we describe the JTT method and

give the asymptotic framework and assumptions to discuss consistency. In Section 3, we describe the main results: consistency of the JTT method and an estimation method after clustering. In Sections 4 and 5, we numerically compare the JTT method with network Lasso, using simulation data and real data. Section 6 concludes the paper. Technical details are provided in the Appendices.

## 2. Preliminaries

We first describe the models and framework of the JTT method. Let  $E_*$  be a set of true edges, and define  $m_*$  as the number of connected components in the graph  $(V, E_*)$  and  $V_i^*$  ( $i \in \{1, \dots, m_*\}$ ) as the vertex set of the  $i$ th connected component. That is,  $m_*$  is the number of true clusters and  $V_1^*, \dots, V_{m_*}^*$  are the nonempty and mutually exclusive sets expressing true clusters, where  $V = \cup_{i=1}^{m_*} V_i^*$ . The  $E_*$  and  $V_i^*$  have the following relationships:

$$\forall k, \ell \in V_i^*, (k, \ell) \in E \implies (k, \ell) \in E_*, \quad \forall (k, \ell) \in E_*, \exists! i \in \{1, \dots, m_*\} \text{ s.t. } k, \ell \in V_i^*.$$

Then, we define the true model as

$$\mathbf{y}_j \sim N_{n_j}(\mathbf{X}_j \boldsymbol{\beta}_j^*, \sigma_*^2 \mathbf{I}_{n_j}) \quad (j \in V), \quad \boldsymbol{\beta}_j^* = \boldsymbol{\xi}_i^* \quad (j \in V_i^*; i \in \{1, \dots, m_*\}),$$

where  $\boldsymbol{\beta}_j^*$  is the  $p$ -dimensional vector of the true regression coefficients for the  $j$ th group,  $\sigma_*^2$  is the true error variance satisfying  $\sigma_* > 0$ , and  $\boldsymbol{\xi}_i^*$  is the  $p$ -dimensional vector of common regression coefficients for the groups in  $V_i^*$ . Thus,  $\boldsymbol{\xi}_i^*$  expresses the relationships among the groups, and the mean structure of the true model is equal within the same cluster. We write  $\boldsymbol{\beta}_k^* = \boldsymbol{\beta}_\ell^* = \boldsymbol{\xi}_{k\ell}^*$  when  $(k, \ell) \in E_*$ . For example, when  $m = 5$  and  $E_* = \{(1, 2), (1, 3), (4, 5)\}$ , we have  $m_* = 2$ ,  $V_1^* = \{1, 2, 3\}$ , and  $V_2^* = \{4, 5\}$ . Furthermore, we have  $\boldsymbol{\beta}_1^* = \boldsymbol{\beta}_2^* = \boldsymbol{\xi}_{12}^* = \boldsymbol{\xi}_1^*$ . For the true model, we define the base model as (1.1), in which  $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_m$  are all different and a candidate model as a model with  $\boldsymbol{\beta}_k = \boldsymbol{\beta}_\ell = \boldsymbol{\xi}_{k\ell}$  only for  $(k, \ell) \in E$ . The candidate model for  $(k, \ell) \in E$  is given by

$$\mathbf{y}_j \sim \begin{cases} N_{n_j}(\mathbf{X}_j \boldsymbol{\xi}_{k\ell}, \sigma_{k\ell}^2 \mathbf{I}_{n_j}) & (j \in \{k, \ell\}) \\ N_{n_j}(\mathbf{X}_j \boldsymbol{\beta}_j, \sigma_{k\ell}^2 \mathbf{I}_{n_j}) & (j \in V_{k\ell}) \end{cases},$$

where  $\sigma_{k\ell}^2$  is an error variance satisfying  $\sigma_{k\ell} > 0$  and  $V_{k\ell} = V \setminus \{k, \ell\}$ . The group-wise OLS estimator of  $\boldsymbol{\beta}_j$  for the candidate model is given by

$$\hat{\boldsymbol{\beta}}_j^{(k\ell)} = \begin{cases} \mathbf{M}_{k\ell}^{-1} \mathbf{X}'_{k\ell} \mathbf{y}_{k\ell} & (j \in \{k, \ell\}) \\ \hat{\boldsymbol{\beta}}_j & (j \in V_{k\ell}) \end{cases}, \quad \mathbf{M}_{k\ell} = \mathbf{X}'_{k\ell} \mathbf{X}_{k\ell}, \quad \mathbf{X}_{k\ell} = \begin{pmatrix} \mathbf{X}_k \\ \mathbf{X}_\ell \end{pmatrix}, \quad \mathbf{y}_{k\ell} = \begin{pmatrix} \mathbf{y}_k \\ \mathbf{y}_\ell \end{pmatrix},$$

where  $\hat{\boldsymbol{\beta}}_j$  is the group-wise OLS estimator of  $\boldsymbol{\beta}_j$  for the base model given in (1.2). Furthermore, we define an unbiased estimator  $s^2$  of the error variance in the base model and projection matrices  $\mathbf{P}_j$  and  $\mathbf{P}_{k\ell}$  as

$$s^2 = \frac{1}{N} \sum_{j=1}^m \mathbf{y}'_j (\mathbf{I}_{n_j} - \mathbf{P}_j) \mathbf{y}_j, \quad \mathbf{P}_j = \mathbf{X}_j \mathbf{M}_j^{-1} \mathbf{X}'_j, \quad \mathbf{P}_{k\ell} = \mathbf{X}_{k\ell} \mathbf{M}_{k\ell}^{-1} \mathbf{X}'_{k\ell}.$$

Then, the  $GC_p$  criteria  $GC_p^{(0)}$  and  $GC_p^{(k\ell)}$  for the base model and the candidate model, respectively, are defined by

$$\begin{aligned} GC_p^{(0)}(\alpha) &= \frac{1}{s^2} \sum_{j=1}^m \mathbf{y}'_j (\mathbf{I}_{n_j} - \mathbf{P}_j) \mathbf{y}_j + \alpha m p, \\ GC_p^{(k\ell)}(\alpha) &= \frac{1}{s^2} \left\{ \mathbf{y}'_{k\ell} (\mathbf{I}_{n_{k\ell}} - \mathbf{P}_{k\ell}) \mathbf{y}_{k\ell} + \sum_{j \in V_{k\ell}} \mathbf{y}'_j (\mathbf{I}_{n_j} - \mathbf{P}_j) \mathbf{y}_j \right\} + \alpha(m-1)p, \end{aligned} \quad (2.1)$$

where  $\alpha (> 0)$  is a parameter adjusting the strength of the penalty against the model fitting and  $n_{k\ell} = n_k + n_\ell$ . The  $GC_p$  criterion expresses a specific criterion by giving the  $\alpha$  value, e.g.,  $\alpha = 2$  expresses the  $C_p$  criterion. Under the models described above, we define the optimal edge set  $\hat{E}(\alpha)$  selected by the JTT method based on the  $GC_p$  criterion as

$$\hat{E}(\alpha) = \{(k, \ell) \in E \mid GC_p^{(k\ell)}(\alpha) \leq GC_p^{(0)}(\alpha)\}. \quad (2.2)$$

When the goodness of a model is evaluated by a model selection criterion, a model with a smaller value can be interpreted as being a better model. In our case, we adopt  $\beta_k = \beta_\ell$  satisfying  $GC_p^{(k\ell)}(\alpha) \leq GC_p^{(0)}(\alpha)$  for  $(k, \ell) \in E$ . Note that the name ‘‘kick-one-out’’ is derived from considering a model from which only one variable is excluded as a candidate model. Following this idea, we have named the proposed method the ‘‘join-two-together’’ method, since a model in which two groups are joined together is considered as a candidate model. Let  $q = \#(E)$ . Then,  $q$  achieves its maximum when  $(V, E)$  is a complete graph. Hence, the JTT method requires a calculation size of  $q = O(m^2)$ , since  $q \leq {}_m C_2 = m(m-1)/2$ . The purpose of this paper is to present a condition on  $\alpha$  such that the JTT method has consistency in the edge selection, i.e., the probability of  $\hat{E}(\alpha) = E_*$  converges to 1.

Next, we provide an asymptotic framework and some assumptions for discussing the consistency. Let  $n_0 = \min_{j \in V} n_j$ . Herein, we consider consistency under the following high-dimensional asymptotic framework:

$$n_0 \rightarrow \infty, \quad \frac{p}{n_0} \rightarrow p_0 \in [0, 1), \quad \frac{m}{n_0} \rightarrow m_0 \in [0, \infty). \quad (2.3)$$

Under this asymptotic framework,  $n_0$  always diverges to infinity. In contrast,  $p$ ,  $m$ , and  $m_*$  can be fixed or diverge to infinity at a speed comparable to or slower than  $n_0$ . We make the following three assumptions in preparation.

Assumption A1.  $E_* \subseteq E$ .

Assumption A2. There exists  $c_1 > 0$  such that

$$n_0^{-1} \min_{(k, \ell) \notin E_*} \lambda_{\min}(\mathbf{M}_k \mathbf{M}_{k\ell}^{-1} \mathbf{M}_\ell) \geq c_1,$$

where  $\lambda_{\min}(\mathbf{A})$  is the minimum eigenvalue of a square matrix  $\mathbf{A}$ .

Assumption A3. There exist  $c_2 > 0$  and  $c_3 \geq 1/2$  such that

$$n_0^{1-c_3} \min_{(k,\ell) \notin E_*} \|\beta_k^* - \beta_\ell^*\|^2 / \sigma_*^2 \geq c_2.$$

Assumption A1 is absolutely necessary for considering consistency. Assumption A2 may seem unusual. For example, the following might be considered a standard assumption.

Assumption A2'. There exists  $d > 0$  such that

$$\min_{j \in V} n_j^{-1} \lambda_{\min}(\mathbf{M}_j) \geq d.$$

Actually, Assumption A2' is a sufficient condition for Assumption A2. Hence, Assumption A2 is a relatively weak assumption. Assumption A3 is a weak assumption for the true parameters. Since it is necessary to determine  $\beta_k \neq \beta_\ell$  for  $(k, \ell) \notin E_*$ , it would be desirable that  $\|\beta_k^* - \beta_\ell^*\|^2$  be sufficiently large. However, Assumption A3 allows  $\|\beta_k^* - \beta_\ell^*\|^2 / \sigma_*^2$  to converge to 0.

Lastly, the asymptotic behavior of a non-centrality parameter is important in model selection. The non-centrality parameter for  $(k, \ell) \in E$  is given by

$$\delta_{k\ell} = \boldsymbol{\eta}_{k\ell}^*{}' (\mathbf{I}_{n_{k\ell}} - \mathbf{P}_{k\ell}) \boldsymbol{\eta}_{k\ell}^* / \sigma_*^2, \quad \boldsymbol{\eta}_{k\ell}^* = \begin{cases} \mathbf{X}_{k\ell} \boldsymbol{\xi}_{k\ell}^* & ((k, \ell) \in E_*) \\ \begin{pmatrix} \mathbf{X}_k \boldsymbol{\beta}_k^* \\ \mathbf{X}_\ell \boldsymbol{\beta}_\ell^* \end{pmatrix} & ((k, \ell) \notin E_*) \end{cases}.$$

Since  $\mathbf{P}_{k\ell} \mathbf{X}_{k\ell} = \mathbf{X}_{k\ell}$ , we know that  $\delta_{k\ell} = 0$  holds when  $(k, \ell) \in E_*$  and  $\delta_{k\ell} > 0$  holds when  $(k, \ell) \notin E_*$ . Regarding  $\delta_{k\ell}$ , we have the following inequality under Assumptions A1–A3 (the proof is given in Appendix A.1):

$$n_0^{-c_3} \delta_{\min} \geq c_1 c_2, \quad \delta_{\min} = \min_{(k,\ell) \notin E_*} \delta_{k\ell}. \quad (2.4)$$

### 3. Main results

#### 3.1. Consistency

We define a class of  $\alpha$  of  $GC_p$  criteria in (2.1) as

$$\mathcal{A} = \left\{ \frac{N}{N-2} + \beta \mid \beta > 0, \frac{\beta p^{1/2}}{m^{1/r_1}} \rightarrow \infty, \frac{\beta p}{n_0^{c_3}} \rightarrow 0 \right\} \quad (r_1 \in \mathbb{N}), \quad (3.1)$$

where  $c_3$  is the constant given in Assumption A3. A  $GC_p$  criterion with  $\alpha$  in  $\mathcal{A}$  is referred to as a high-dimensionality-adjusted consistent  $GC_p$  ( $HCGC_p$ ) criterion. (This name is based on, e.g., Yanagihara, 2016 and Oda & Yanagihara, 2020.) We have the following theorem on the JTT method based on the  $HCGC_p$  criterion (the proof is given in Appendix A.2).

**Theorem 1.** *Suppose that Assumptions A1–A3 hold. Then, under the asymptotic framework in (2.3), we have*

$$\forall \alpha \in \mathcal{A}, \mathbb{P}(\hat{E}(\alpha) = E_*) \rightarrow 1.$$

Moreover, the convergence order is given by

$$\mathbb{P}(\hat{E}(\alpha) = E_*) = 1 - \begin{cases} O(m^2 \beta^{-2r_1} p^{-r_1} + n_0^{2-c_3 r_2} + m^{2-r_2} n_0^{-r_2}) & (c_3 \geq 1) \\ O(m^2 \beta^{-2r_1} p^{-r_1} + n_0^{2+r_2-2c_3 r_2}) & (1/2 \leq c_3 < 1) \end{cases},$$

where  $r_2$  is a natural number satisfying  $r_2 > 2$ .

Theorem 1 guarantees that the JTT method based on the  $HCGC_p$  criterion has consistency in the edge selection. To use this method in practice, we need to determine a specific value of  $\alpha$  ( $\in \mathcal{A}$ ). For example,  $\alpha = 2$  expressing the  $C_p$  criterion and  $\alpha = \log n$  corresponding to the Bayesian information criterion (Schwarz, 1978) are often used, but they do not belong to  $\mathcal{A}$  (e.g.,  $\alpha = 2$  belongs to  $\mathcal{A}$  when  $m$  is fixed and  $p = O(\log n_0)$ , and  $\alpha = \log n$  belongs to  $\mathcal{A}$  when  $n = mn_0$  and  $p$  and  $m$  are fixed). As a specific  $\alpha$  value satisfying the conditions of  $\mathcal{A}$ , we propose

$$\hat{\alpha} = \frac{N}{N-2} + \hat{\beta}, \quad \hat{\beta} = B \cdot \frac{m^{1/4} \log n_0}{\sqrt{p}}, \quad B = \frac{N \sqrt{N+p-2}}{(N-2) \sqrt{N-4}}. \quad (3.2)$$

If  $r_1 \geq 4$  and  $c_3 > 3/4$ , then  $\hat{\alpha} \in \mathcal{A}$ . Hence, the probability that  $\hat{E}(\hat{\alpha})$  is equal to  $E_*$  converges to 1. Notice that  $B$  in  $\hat{\beta}$  does not affect the consistency result because  $B$  is of constant order. Based on Yanagihara (2016),  $B$  is incorporated to standardize  $HCGC_p^{(k\ell)}(\hat{\alpha}) - HCGC_p^{(0)}(\hat{\alpha})$  for  $(k, \ell) \in E_*$  (the details are given in Appendix A.3).

We can obtain another condition for the consistency of the JTT method based on a  $GC_p$  criterion as the following theorem (the proof is given in Appendix A.4).

**Theorem 2.** *Suppose that Assumptions A1–A3 hold. Furthermore, we define a class of  $\alpha$  in  $GC_p$  criteria as*

$$\check{\mathcal{A}} = \left\{ \alpha > \frac{2}{1-r} \mid \frac{\alpha p}{\log m} \rightarrow \infty, \frac{\alpha p}{n_0^{c_3}} \rightarrow 0 \right\} \quad (r \in (0, 1)).$$

Then, under the asymptotic framework in (2.3), we have

$$\forall \alpha \in \check{\mathcal{A}}, \mathbb{P}(\hat{E}(\alpha) = E_*) \rightarrow 1.$$

Moreover, the convergence order is given by

$$\begin{aligned} & \mathbb{P}(\hat{E}(\alpha) = E_*) \\ &= 1 - O\left(\exp[-\alpha h p \{(1-r_1) - 1/\alpha\}] + m^2 \left\{ \exp(-r_1 N/4) + \exp(-c_1 c_2 n_0^{c_3}/8) + \exp(-hr_2 N) \right\}\right), \end{aligned}$$

where  $h = (1 - \log 2)/2$ ,  $r_1 \in (0, 1)$ , and  $r_2 \in [1, \infty)$ .

Although the class of  $\alpha$  given in Theorem 2 looks like a relaxed version of  $\mathcal{A}$  in (3.1), it is difficult to determine a specific  $\alpha$  value due to the condition  $\alpha > 2/(1-r)$ . For example, consider  $\check{\alpha}$  given by

$$\check{\alpha} = 2 + \frac{m^{1/4} \log n_0}{\sqrt{p}}.$$

If  $p/n_0^{c_3} \rightarrow 0$ , then the probability that  $\hat{E}(\check{\alpha})$  is equal to  $E_*$  converges to 1.

We have already discussed a condition that guarantees consistency of the JTT method. Here, we will discuss a condition under which it is not consistent. The condition for inconsistency of the JTT method is given by the following theorem (the proof is given in Appendix A.5).

**Theorem 3.** *Suppose that Assumptions A1–A3 hold. Then given the two conditions*

$$C1 : \begin{cases} \alpha \not\rightarrow \infty & (p: \text{fixed}) \\ \liminf \alpha < 1 & (p \rightarrow \infty) \end{cases}, \quad C2 : \begin{cases} \alpha p / \delta_{\min} \rightarrow \infty & (\delta_{\min} / p \rightarrow \infty) \\ \limsup \alpha > 1 + c_4 & (\delta_{\min} / p \rightarrow c_4 \in [0, \infty)) \end{cases},$$

under the asymptotic framework in (2.3), for  $\alpha$  satisfying either C1 or C2, we have

$$P(\hat{E}(\alpha) = E_*) \not\rightarrow 1.$$

For example,  $\alpha = 2$  guarantees inconsistency when  $p$  is fixed or there exists  $c_4 \in [0, 1)$  (e.g.,  $\delta_{\min} = d_1 n_0$  and  $p = (d_1 + \epsilon) n_0$  for  $d_1 \in (0, 1)$  and  $\epsilon \in (0, 1 - d_1)$ ), and  $\alpha = \log n$  guarantees inconsistency when there exists  $c_4 \in [0, \infty)$  (e.g.,  $\delta_{\min} = d_2 n_0$  and  $p = d_3 n_0$  for  $d_2 > 0$  and  $d_3 \in (0, 1)$ ).

### 3.2. Post-selection estimation

In this section, we discuss an estimation method for  $\beta_1, \dots, \beta_m$  after obtaining the optimal edge set  $\hat{E}(\alpha)$  selected by the JTT method. One of simplest methods is the cluster-wise OLS method. Let  $\hat{m}$  be the number of connected components of the graph  $(V, \hat{E}(\alpha))$  and  $\hat{V}_i$  ( $i \in \{1, \dots, \hat{m}\}$ ) be a vertex set of the  $i$ th connected component, where  $\hat{V}_1, \dots, \hat{V}_{\hat{m}}$  are nonempty and mutually exclusive sets satisfying  $V = \cup_{i=1}^{\hat{m}} \hat{V}_i$ . This notation means that the JTT method produced  $\hat{m}$  clusters and  $\hat{V}_i$  expresses the  $i$ th cluster. Let  $\tilde{\mathbf{y}}_i$  and  $\tilde{\mathbf{X}}_i$  ( $i \in \{1, \dots, \hat{m}\}$ ) be an  $\tilde{n}_i$ -dimensional vector and  $\tilde{n}_i \times p$  matrix constructed by vertically stacking, respectively, the vectors  $\mathbf{y}_j$  and matrices  $\mathbf{X}_j$  for  $j \in \hat{V}_i$ , where  $\tilde{n}_i = \sum_{j \in \hat{V}_i} n_j$ . Then, the cluster-wise OLS estimator of the regression coefficient vector for the  $i$ th cluster is given by

$$\hat{\xi}_i = (\tilde{\mathbf{X}}_i' \tilde{\mathbf{X}}_i)^{-1} \tilde{\mathbf{X}}_i' \tilde{\mathbf{y}}_i \quad (i \in \{1, \dots, \hat{m}\}).$$

If  $\hat{E}(\alpha) = E_*$ , then  $\hat{\xi}_i$  is a good estimator of  $\xi_i^*$ . However, its estimation accuracy may be low when the sample size within the cluster is small.

We try to improve the estimation accuracy by estimating with information of other clusters. Specifically, in order to maintain the simplicity of the JTT method, a method with low calculation cost is desirable. To achieve this, we employ a penalized estimation method with a weighted average of connected clusters. Let  $\hat{F}$  be an edge set for the clusters based on the original edge set  $E$ . Specifically,  $\hat{F}$

is the edge set for the vertex set  $\{1, \dots, \hat{m}\}$  and  $(k, \ell) \in \hat{F} \Rightarrow k < \ell$ . Given  $\hat{F}$ , we define a weighted average as

$$\mathbf{b}_i = \frac{\sum_{(i,\ell) \in \hat{F}} w_{i\ell} \hat{\boldsymbol{\xi}}_\ell + \sum_{(k,i) \in \hat{F}} w_{ik} \hat{\boldsymbol{\xi}}_k}{\sum_{(i,\ell) \in \hat{F}} w_{i\ell} + \sum_{(k,i) \in \hat{F}} w_{ik}}, \quad w_{k\ell} = \|\hat{\boldsymbol{\xi}}_k - \hat{\boldsymbol{\xi}}_\ell\|^{-1}.$$

Then, the estimator for the  $i$ th cluster is defined by

$$\hat{\boldsymbol{\xi}}_i(\lambda) = \arg \min_{\boldsymbol{\xi}_i} \left\{ \|\tilde{\mathbf{y}}_i - \tilde{\mathbf{X}}_i \boldsymbol{\xi}_i\|^2 + \lambda \|\boldsymbol{\xi}_i - \mathbf{b}_i\|^2 \right\} = \left( \tilde{\mathbf{X}}_i' \tilde{\mathbf{X}}_i + \lambda \mathbf{I}_p \right)^{-1} \left( \tilde{\mathbf{X}}_i' \tilde{\mathbf{y}}_i + \lambda \mathbf{b}_i \right),$$

where  $\lambda > 0$  is a tuning parameter. By selecting  $\lambda$  appropriately, we can expect to improve the estimation accuracy. Here, we consider selecting  $\lambda$  based on prediction accuracy, i.e., predictive mean square error (PMSE). Let  $\widehat{\tilde{\mathbf{y}}}_i(\lambda)$  be a vector of fitted values obtained from  $\hat{\boldsymbol{\xi}}_i(\lambda)$ , i.e.,  $\widehat{\tilde{\mathbf{y}}}_i(\lambda) = \tilde{\mathbf{X}}_i \hat{\boldsymbol{\xi}}_i(\lambda)$ . Based on Fujikoshi & Satoh (1997), the modified  $C_p$  ( $MC_p$ ) criterion is given by

$$MC_p(\lambda | i) = \frac{1}{s_i^2} \|\tilde{\mathbf{y}}_i - \widehat{\tilde{\mathbf{y}}}_i(\lambda)\|^2 + \frac{2(\tilde{n}_i - p)}{\tilde{n}_i - p - 2} \text{tr } \mathbf{H}_i(\lambda),$$

$$s_i^2 = \frac{\|\tilde{\mathbf{y}}_i - \widehat{\tilde{\mathbf{y}}}_i(0)\|^2}{\tilde{n}_i - p}, \quad \mathbf{H}_i(\lambda) = \left( \tilde{\mathbf{X}}_i' \tilde{\mathbf{X}}_i + \lambda \mathbf{I}_p \right)^{-1} \tilde{\mathbf{X}}_i' \tilde{\mathbf{X}}_i,$$

and we define the optimal tuning parameter for the  $i$ th cluster as

$$\hat{\lambda}_i = \arg \min_{\lambda} MC_p(\lambda | i).$$

Notice that  $MC_p(\lambda | i)$  is an unbiased estimator of PMSE for  $\widehat{\tilde{\mathbf{y}}}_i(\lambda)$  by regarding  $w_{i\ell}$  as a constant. Hence, we can expect to improve the prediction accuracy by selecting  $\lambda$  minimizing  $MC_p(\lambda | i)$  (actually, since  $w_{i\ell}$  depends on  $\tilde{\mathbf{y}}_i$ ,  $MC_p(\lambda | i)$  is a naive estimator). On the other hand, because obtaining  $\hat{\lambda}_i$  requires a numerical search and the inverse matrix in  $MC_p(\lambda | i)$  depends on  $\lambda$ , this estimation method seems not to have a low calculation cost. Fortunately, we can avoid calculating the inverse matrix by using a singular value decomposition of  $\tilde{\mathbf{X}}_i$ . Note that  $\tilde{\mathbf{X}}_i$  can be decomposed as

$$\tilde{\mathbf{X}}_i = \mathbf{U}_i \mathbf{D}_i^{1/2} \mathbf{V}_i', \quad \mathbf{D}_i = \text{diag}(d_{i1}, \dots, d_{ip}),$$

where  $\mathbf{U}_i$  is an  $\tilde{n}_i \times p$  matrix,  $\mathbf{V}_i$  is an orthogonal matrix of order  $p$ , and the diagonal elements of  $\mathbf{D}_i$  satisfy  $d_{i1} \geq \dots \geq d_{ip} > 0$ . Then, the  $MC_p$  criterion can be rewritten as follows (e.g., Yanagihara, 2018):

$$MC_p(\lambda | i) = \tilde{n}_i - p + \frac{1}{s_i^2} \sum_{j=1}^p \left( \frac{\lambda}{d_{ij} + \lambda} \right)^2 (z_{ij} - d_{ij} r_{ij})^2 + \frac{2(\tilde{n}_i - p)}{\tilde{n}_i - p - 2} \left( p - \sum_{j=1}^p \frac{\lambda}{d_{ij} + \lambda} \right),$$

where  $z_{ij}$  and  $r_{ij}$  are the  $j$ th elements of  $\mathbf{U}_i' \tilde{\mathbf{y}}_i$  and  $\mathbf{D}_i^{-1/2} \mathbf{V}_i' \mathbf{b}_i$ , respectively. Since this expression eliminates the inverse matrix, searching for  $\hat{\lambda}_i$  numerical is simpler.

#### 4. Simulation

In this section, we evaluate the performance of the JTT method proposed in this paper through Monte Carlo simulation with 1000 iterations. We also compare it with network Lasso. The numerical calculation programs are executed in R (ver. 4.5.0; R Core Team, 2025) on a computer running the Windows 11 Pro operating system with an AMD EPYC TM 7763 processor and 128 GB of RAM. Network Lasso is implemented as R package GGFL (ver. 1.0.2; Ohishi, 2025a), which employs BCDM (Ohishi *et al.*, 2025). The JTT method is available via R package JTT (ver. 0.1.0; Ohishi, 2025b). Note that although the JTT package includes some C++ code, the numerical calculation is conducted without the C++ code for a fair comparison of runtime (the GGFL package consists only of R code).

We first describe the setting of the simulation model. Let  $(V, E)$ , which expresses the relationships among the  $m$  groups, be a complete graph, and define sets  $V_i^*$  ( $i \in \{1, \dots, m_*\}$ ), which express the true clusters, for  $m_*/m \in \{0.3, 0.6\}$  ( $m \in \{20, 50\}$ ), where  $V = \{1, \dots, m\}$ . Although the definition of each  $V_i^*$  is omitted, they are available in the JTT package and the definition for  $m = 20$  is the same as that in Ohishi *et al.* (2021). Then, the simulation model is defined by

$$\mathbf{y}_j \sim N_{n_0}(\mathbf{X}_j \boldsymbol{\beta}_j^*, \mathbf{I}_{n_0}) \quad (j \in V), \quad \boldsymbol{\beta}_j^* = \boldsymbol{\xi}_i^* = \nu i \mathbf{1}_p \quad (j \in V_i^*; i \in \{1, \dots, m_*\}),$$

where  $\mathbf{X}_j = (\mathbf{1}_{n_0}, \mathbf{Z}_j \boldsymbol{\Psi}(0.5)^{1/2})$ ,  $\mathbf{Z}_j$  is an  $n_0 \times (p-1)$  matrix with elements identically and independently distributed according to  $U(-1, 1)$ ,  $\boldsymbol{\Psi}(\rho)$  is a matrix of order  $p-1$  with  $(i, j)$ th elements  $\rho^{|i-j|}$ ,  $\nu$  is a constant adjusting the signal-to-noise ratio (SNR), and  $\mathbf{1}_p$  is a  $p$ -dimensional vector of ones. Here, SNR is defined by

$$\text{SNR} = \frac{1}{\#(F_*)} \sum_{(k, \ell) \in F_*} \frac{\text{Var}[\mathbf{x}'(\boldsymbol{\xi}_k^* - \boldsymbol{\xi}_\ell^*)]}{(p-1)\sigma_*^2} = \frac{1}{3(p-1)\#(F_*)} \sum_{(k, \ell) \in F_*} (\boldsymbol{\theta}_k^* - \boldsymbol{\theta}_\ell^*)' \boldsymbol{\Psi}(0.5) (\boldsymbol{\theta}_k^* - \boldsymbol{\theta}_\ell^*),$$

where  $F_*$  is the edge set for the vertex set  $\{1, \dots, m_*\}$  of the true clusters,  $\mathbf{x} = (1, \boldsymbol{\Psi}(0.5)^{1/2} \mathbf{z})'$ ,  $\mathbf{z}$  is a  $(p-1)$ -dimensional vector with elements identically and independently distributed according to  $U(-1, 1)$ ,  $\sigma_*^2 = 1$ , and  $\boldsymbol{\theta}_k^*$  is the vector obtained by removing the first element from  $\boldsymbol{\xi}_k^*$ . As described in Assumption A3, the difference of the true regression coefficient vectors for two groups is important for guaranteeing the consistency of the JTT method. Hence, SNR is defined based on this difference. Furthermore, to slow down the increase in SNR as  $p$  increases, SNR is standardized by  $p-1$ . In this simulation,  $\nu$  is defined to satisfy  $\text{SNR} = 3$ .

Under the setting described above, we evaluate clustering accuracy, mean square error (MSE), and runtime. Clustering accuracy here means the rate (%) at which the true clusters are selected over 1000 iterations. MSEs are defined for a vector of fitted values  $\mathbf{y}^\dagger = (\mathbf{y}_1^\dagger, \dots, \mathbf{y}_m^\dagger)'$  and an estimator of regression coefficient vector  $\boldsymbol{\beta}^\dagger = (\boldsymbol{\beta}_1^\dagger, \dots, \boldsymbol{\beta}_m^\dagger)'$ , respectively, as

$$\text{MSE}_f[\mathbf{y}^\dagger] = \frac{1}{n} \mathbb{E} \left[ \sum_{j=1}^m \|\mathbf{X}_j \boldsymbol{\beta}_j^* - \mathbf{y}_j^\dagger\|^2 \right], \quad \text{MSE}_c[\boldsymbol{\beta}^\dagger] = \frac{1}{mp} \mathbb{E} \left[ \sum_{j=1}^m \|\boldsymbol{\beta}_j^* - \boldsymbol{\beta}_j^\dagger\|^2 \right],$$

where expectation is evaluated by Monte Carlo simulation with 1000 iterations. In the simulation, MSEs for  $\mathbf{y}_j^\dagger = \mathbf{X}_j\boldsymbol{\beta}_j^\dagger$  and  $\boldsymbol{\beta}_j^\dagger = \hat{\boldsymbol{\beta}}_j$  are given by  $\text{MSE}_f[\mathbf{y}_j^\dagger] = mp/n = p/n_0$  and  $\text{MSE}_c[\boldsymbol{\beta}_j^\dagger] = \sum_{j=1}^m \text{tr}(\mathbf{M}_j^{-1})/mp$ , respectively, where  $\hat{\boldsymbol{\beta}}_j$  is the group-wise OLS estimator of  $\boldsymbol{\beta}_j$  given in (1.2). The methods used in this simulation are the JTT method and network Lasso. The  $HCGC_p$  criterion with  $\hat{\alpha}$  in (3.2) is used in the JTT method. Furthermore, the cluster-wise OLS method and the penalized estimation method described in Section 3.2 are applied as an estimation method after clustering, denoted as JTT1 and JTT2, respectively. Network Lasso is implemented with the GGFL package under default settings. The penalty weights are thus given by  $w_{k\ell} = \|\hat{\boldsymbol{\beta}}_k - \hat{\boldsymbol{\beta}}_\ell\|^{-1}$ , and tuning parameter  $\lambda$  is selected based on minimizing the following extended GCV (EGCV) criterion (Ohishi *et al.*, 2020):

$$\text{EGCV}(\lambda \mid \alpha) = \frac{\sum_{j=1}^m \|\mathbf{y}_j - \hat{\mathbf{y}}_{\lambda,j}\|^2}{(1 - \text{df}_\lambda/n)^\alpha},$$

where  $\hat{\mathbf{y}}_{\lambda,j}$  is the vector of fitted values obtained from the network Lasso estimator,  $\text{df}_\lambda$  is the number of unique parameters, and  $\alpha = \log n$ . We denote network Lasso as NL in the presented results.

Table 1. Clustering accuracy (%) for fixed  $p$ 

		$p = 20$				$p = 40$			
		$m_*/m = 0.3$		$m_*/m = 0.6$		$m_*/m = 0.3$		$m_*/m = 0.6$	
$m$	$n_0$	JTT	NL	JTT	NL	JTT	NL	JTT	NL
20	50	<b>100.0</b>	0.0	0.0	0.0	<b>100.0</b>	0.0	<b>0.3</b>	0.0
	100	<b>100.0</b>	30.3	<b>95.0</b>	10.8	<b>100.0</b>	2.6	<b>100.0</b>	1.4
	200	<b>100.0</b>	76.9	<b>100.0</b>	35.6	<b>100.0</b>	71.8	<b>100.0</b>	45.6
	500	<b>100.0</b>	93.6	<b>100.0</b>	80.5	<b>100.0</b>	97.9	<b>100.0</b>	94.0
	1000	<b>100.0</b>	99.2	<b>100.0</b>	98.0	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	99.1
50	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	100	0.0	0.0	0.0	0.0	<b>1.1</b>	0.0	0.0	0.0
	200	<b>97.7</b>	0.1	0.0	0.0	<b>100.0</b>	0.0	0.0	0.0
	500	<b>100.0</b>	8.7	0.0	<b>0.6</b>	<b>100.0</b>	9.1	<b>89.7</b>	2.5
	1000	<b>100.0</b>	23.0	<b>100.0</b>	8.2	<b>100.0</b>	30.6	<b>100.0</b>	18.2

Tables 1 and 2 summarize clustering accuracy for fixed  $p$  and  $p/n_0$ , respectively. In these tables, bold font indicates the higher clustering accuracy. As shown in Table 1, for fixed  $p$ , the JTT method performed better than network Lasso in clustering, with its accuracy reaching 100% at a much smaller  $n_0$ . For  $m = 50$  and  $m_*/m = 0.6$ , neither method could select the true clusters for smaller  $n_0$ . Actually,  $\min_{(k,\ell) \in E_*} \|\boldsymbol{\xi}_k^* - \boldsymbol{\xi}_\ell^*\|^2/\sigma_*^2$  is extremely small, making the determination of the true clusters very difficult. In this situation, the JTT method unified all groups into one cluster in most cases. Network Lasso behaved differently. For example, for  $m = 50$ ,  $m_*/m = 0.6$ , and  $n_0 = 100, 200$ , network Lasso determined that every group defines its own cluster. Although network Lasso could not determine the

true clusters like the JTT method, it performed a conservative selection by selecting the smallest  $\lambda$ . However, the JTT method was able to select the true clusters for sufficiently large  $n_0$ . One the other

Table 2. Clustering accuracy (%) for fixed  $p/n_0$

		$p/n_0 = 0.4$				$p/n_0 = 0.8$			
		$m_*/m = 0.3$		$m_*/m = 0.6$		$m_*/m = 0.3$		$m_*/m = 0.6$	
$m$	$n_0$	JTT	NL	JTT	NL	JTT	NL	JTT	NL
20	50	<b>100.0</b>	0.6	0.0	0.0	<b>100.0</b>	0.0	<b>1.3</b>	0.0
	100	<b>100.0</b>	1.7	<b>100.0</b>	1.3	<b>100.0</b>	0.0	<b>99.9</b>	0.0
	200	<b>100.0</b>	14.2	<b>100.0</b>	22.5	<b>100.0</b>	0.0	<b>100.0</b>	0.0
	500	<b>100.0</b>	72.8	<b>100.0</b>	92.8	<b>100.0</b>	0.0	<b>100.0</b>	0.0
	1000	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	4.1	<b>100.0</b>	20.3
50	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	100	<b>1.4</b>	0.0	0.0	0.0	<b>10.1</b>	0.0	0.0	0.0
	200	<b>100.0</b>	0.0	0.0	0.0	<b>100.0</b>	0.0	0.0	0.0
	500	<b>100.0</b>	0.0	<b>100.0</b>	0.0	<b>100.0</b>	0.0	<b>100.0</b>	0.0
	1000	<b>100.0</b>	7.8	<b>100.0</b>	0.0	<b>100.0</b>	0.0	<b>100.0</b>	0.0

hand, when  $p$  increases with  $n_0$ , as shown in Table 2, the JTT method maintained high clustering accuracy, whereas network Lasso was not able to select the true clusters, particularly for large  $p$  or  $m$ .

Table 3. Relative MSE (%) for  $p = 20$

		$m_*/m = 0.3$						$m_*/m = 0.6$					
		RMSE <sub>f</sub>			RMSE <sub>c</sub>			RMSE <sub>f</sub>			RMSE <sub>c</sub>		
$m$	$n_0$	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL
20	50	30.0	<b>24.6</b>	286.7	20.6	<b>13.8</b>	58.6	921.0	921.7	104.0	176.2	162.4	<b>98.7</b>
	100	30.3	<b>26.1</b>	144.4	25.7	<b>19.1</b>	37.3	62.0	<b>46.9</b>	159.7	55.7	<b>31.9</b>	58.8
	200	30.9	<b>28.0</b>	87.9	28.3	<b>23.0</b>	30.2	60.6	<b>48.6</b>	129.8	58.0	<b>38.1</b>	42.0
	500	32.4	<b>30.7</b>	56.4	29.8	<b>26.4</b>	27.3	62.3	<b>53.9</b>	93.9	59.7	45.1	<b>39.7</b>
	1000	34.9	<b>33.8</b>	46.8	30.7	28.5	<b>27.9</b>	64.5	<b>58.3</b>	78.3	60.3	49.1	<b>41.9</b>
50	50	6159.5	6159.5	3708.9	857.4	857.4	542.7	6600.9	6600.9	3561.4	866.7	866.6	507.8
	100	2259.5	2260.5	146.0	429.9	425.0	<b>97.4</b>	12752.5	12752.5	100.1	2324.6	2324.6	<b>99.6</b>
	200	31.0	<b>25.6</b>	287.1	28.0	<b>19.7</b>	69.8	21107.2	21107.5	100.7	4314.9	4314.4	<b>99.9</b>
	500	29.9	<b>26.0</b>	166.1	29.3	<b>23.0</b>	45.8	297.5	290.0	180.6	101.0	87.7	<b>58.0</b>
	1000	29.9	<b>26.8</b>	110.7	29.5	<b>24.7</b>	35.6	65.1	<b>56.4</b>	156.8	60.7	<b>45.7</b>	47.7

Tables 3–6 summarize MSE for both the fitted values and estimator of regression coefficients of

which values are relative MSE (RMSE; %) based on MSE obtained from the group-wise OLS estimator

$$\text{RMSE}_f[\mathbf{y}^\dagger] = 100 \times \frac{\text{MSE}_f[\mathbf{y}^\dagger]}{mp/n}, \quad \text{RMSE}_c[\boldsymbol{\beta}^\dagger] = 100 \times \frac{\text{MSE}_c[\boldsymbol{\beta}^\dagger]}{\sum_{j=1}^m \text{tr}(\mathbf{M}_j^{-1})/mp}.$$

An RMSE value smaller than 100 means that the method is superior to the group-wise OLS method in estimation accuracy. In these tables, bold font indicates the minimum value, if any value is less than 100. Notice that under the true clusters, MSE for the fitted values of JTT1 is  $m_*p/n$ . Hence, RMSE for the fitted values of JTT1 is  $m_*/m$  if the clustering accuracy of the JTT method is 100% (since the values in the tables are approximations from Monte Carlo simulation, they do not coincide exactly with  $m_*/m$  even when the clustering accuracy is 100%). In the tables, we can see that JTT2 had the best estimation accuracy for both fitted values and regression coefficients for most cases. Recall that JTT2 employs a simple penalized estimation method. Nevertheless, it improved MSE compared to JTT1 for most cases. Although we can see that  $\text{RMSE}_f$  of JTT1 when  $p$  is fixed got worse as  $n_0$  increased, this is a problem not of the estimation accuracy but the approximation accuracy of Monte Carlo simulation, because  $\text{MSE}_f$  gets smaller as  $n_0$  increases. The same behavior was not observed for  $p/n_0$  fixed, because  $\text{MSE}_f$  remained constant as  $n_0$  increased. Moreover, when clustering accuracy was extremely low, i.e., it was difficult to determine the true clusters, the JTT method and network Lasso showed different tendencies. For instance, for  $m = 50$ ,  $m_*/m = 0.6$ , and  $n_0 = 100, 200$  in

Table 4. Relative MSE (%) for  $p = 40$ 

		$m_*/m = 0.3$						$m_*/m = 0.6$					
		RMSE <sub>f</sub>			RMSE <sub>c</sub>			RMSE <sub>f</sub>			RMSE <sub>c</sub>		
$m$	$n_0$	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL
20	50	29.9	<b>23.6</b>	4919.4	7.7	<b>4.7</b>	246.0	1132.0	1142.8	5248.8	77.3	<b>63.9</b>	266.5
	100	30.0	<b>25.4</b>	212.6	20.7	<b>14.8</b>	41.2	59.8	<b>43.3</b>	211.1	50.4	<b>26.0</b>	67.2
	200	30.3	<b>27.1</b>	102.6	25.8	<b>20.6</b>	29.4	60.2	<b>47.1</b>	149.2	55.9	<b>35.0</b>	42.4
	500	31.2	<b>29.4</b>	56.3	28.6	<b>25.0</b>	25.7	61.1	<b>51.8</b>	89.8	58.6	42.9	<b>36.4</b>
	1000	32.4	<b>31.3</b>	43.5	29.6	27.2	<b>26.3</b>	62.2	<b>55.4</b>	70.1	59.4	47.5	<b>39.1</b>
50	50	5843.6	5844.1	6041.4	248.1	247.9	256.5	6018.1	6018.1	6018.1	264.0	264.0	264.0
	100	169.3	164.8	199.6	38.7	<b>31.2</b>	101.6	12353.1	12353.2	112.4	1649.7	1649.5	101.6
	200	30.0	<b>24.0</b>	343.9	26.0	<b>17.6</b>	70.8	11738.4	11743.1	100.2	2101.1	2095.6	<b>99.7</b>
	500	30.0	<b>25.7</b>	188.7	28.4	<b>21.9</b>	45.7	62.8	<b>50.4</b>	225.2	59.0	<b>38.8</b>	63.0
	1000	30.0	<b>26.6</b>	115.6	29.2	<b>24.1</b>	33.3	62.5	<b>52.9</b>	161.6	59.9	<b>43.7</b>	44.6

Table 4, RMSE of the JTT method is extremely large, whereas that of network Lasso is close to 100. This difference of RMSE can in turn be explained by the difference of clustering behaviors. Since the JTT method unifies all groups into one cluster, MSE gets much worse (in this case, MSEs of JTT1

and JTT2 are equal). In contrast, when network Lasso selects the smallest  $\lambda$  and no groups are joined together, the estimator is close to the group-wise OLS estimator and so RMSE is close to 100.

Table 5. Relative MSE (%) for  $p/n_0 = 0.4$

$m$	$n_0$	$m_*/m = 0.3$						$m_*/m = 0.6$					
		RMSE <sub>f</sub>			RMSE <sub>c</sub>			RMSE <sub>f</sub>			RMSE <sub>c</sub>		
		JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL
20	50	30.1	<b>24.4</b>	285.6	20.3	<b>13.3</b>	49.0	536.0	532.9	118.0	110.4	<b>91.2</b>	95.1
	100	29.9	<b>25.3</b>	216.7	20.5	<b>14.6</b>	39.8	59.8	<b>43.2</b>	199.4	50.7	<b>25.9</b>	69.3
	200	29.9	<b>26.4</b>	151.7	20.5	<b>16.1</b>	32.3	59.8	<b>45.5</b>	207.2	50.6	<b>29.2</b>	46.3
	500	29.9	<b>27.8</b>	84.8	20.6	<b>17.6</b>	23.8	59.9	<b>49.2</b>	122.0	50.9	<b>34.3</b>	37.9
	1000	30.0	<b>28.6</b>	52.2	20.6	<b>18.7</b>	20.3	59.9	<b>51.8</b>	76.6	50.9	37.7	<b>33.9</b>
50	50	5875.2	5875.3	5043.5	845.6	845.5	738.7	5846.1	5846.1	2241.9	867.8	867.8	384.7
	100	220.2	216.2	265.6	45.5	<b>38.2</b>	101.5	12518.2	12518.2	<b>99.9</b>	1609.1	1609.1	<b>99.9</b>
	200	30.0	<b>23.7</b>	571.7	21.3	<b>13.8</b>	88.0	2906.6	2909.7	<b>99.9</b>	407.6	395.2	<b>99.9</b>
	500	30.0	<b>25.2</b>	295.3	21.3	<b>15.7</b>	47.0	60.0	<b>46.0</b>	259.1	51.6	<b>30.3</b>	81.4
	1000	30.0	<b>26.2</b>	338.0	21.3	<b>16.9</b>	55.8	60.0	<b>48.8</b>	272.5	51.6	<b>34.0</b>	52.6

Table 6. Relative MSE (%) for  $p/n_0 = 0.8$

$m$	$n_0$	$m_*/m = 0.3$						$m_*/m = 0.6$					
		RMSE <sub>f</sub>			RMSE <sub>c</sub>			RMSE <sub>f</sub>			RMSE <sub>c</sub>		
		JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL
20	50	30.0	<b>23.8</b>	3348.4	7.6	<b>4.8</b>	164.1	586.9	586.6	5105.2	51.7	<b>38.9</b>	242.7
	100	30.0	<b>24.9</b>	7885.0	7.9	<b>5.4</b>	372.9	60.0	<b>42.6</b>	11216.0	36.2	<b>11.9</b>	546.0
	200	29.9	<b>26.2</b>	897.8	8.0	<b>6.1</b>	65.2	60.0	<b>44.9</b>	20216.5	36.1	<b>14.6</b>	1028.5
	500	30.0	<b>27.7</b>	272.1	8.1	<b>6.8</b>	25.5	60.0	<b>48.2</b>	53670.8	36.2	<b>17.6</b>	2481.2
	1000	30.0	<b>28.5</b>	144.6	8.1	<b>7.3</b>	16.8	60.0	<b>50.7</b>	73500.7	36.4	<b>21.1</b>	3718.0
50	50	5989.4	5989.8	6053.9	245.9	245.7	245.7	5997.2	5997.2	5934.3	255.1	255.1	250.3
	100	134.6	129.4	12369.6	14.6	<b>9.8</b>	502.3	12436.0	12436.2	12531.6	530.7	530.6	534.9
	200	30.0	<b>23.4</b>	20950.9	9.4	<b>5.3</b>	940.8	2652.1	2662.7	24446.2	139.1	127.2	1055.8
	500	30.0	<b>25.0</b>	57578.4	9.5	<b>6.2</b>	2489.3	60.0	<b>45.2</b>	61407.8	38.2	<b>15.5</b>	2607.4
	1000	30.0	<b>26.0</b>	565.5	9.6	<b>6.8</b>	43.2	60.0	<b>47.8</b>	122132.5	38.1	<b>18.1</b>	5155.1

Tables 7 and 8 summarize runtime, with bold font indicating the smallest value. As shown, JTT1 was much faster than network Lasso. Furthermore, we can see that the penalized estimation in JTT2 does not affect runtime much. Since the JTT method requires repeated calculation of the inverse of a

Table 7. Runtime (sec.) for fixed  $p$ 

		$p = 20$						$p = 40$					
		$m_*/m = 0.3$			$m_*/m = 0.6$			$m_*/m = 0.3$			$m_*/m = 0.6$		
$m$	$n_0$	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL
20	50	<b>0.1</b>	0.1	21.0	<b>0.1</b>	0.1	22.7	<b>0.1</b>	0.2	31.8	<b>0.1</b>	0.1	42.0
	100	<b>0.1</b>	0.1	16.1	<b>0.1</b>	0.1	19.1	<b>0.1</b>	0.1	21.4	<b>0.1</b>	0.1	30.2
	200	<b>0.1</b>	0.1	14.1	<b>0.1</b>	0.1	17.2	<b>0.1</b>	0.2	16.6	<b>0.1</b>	0.2	21.4
	500	<b>0.1</b>	0.1	12.4	<b>0.1</b>	0.2	14.6	<b>0.1</b>	0.3	14.3	<b>0.1</b>	0.3	17.3
	1000	<b>0.1</b>	0.2	11.9	<b>0.1</b>	0.3	13.8	<b>0.3</b>	0.4	13.5	<b>0.2</b>	0.5	16.2
50	50	<b>0.2</b>	0.2	290.6	<b>0.2</b>	0.2	344.2	0.3	<b>0.3</b>	496.8	0.3	<b>0.3</b>	532.8
	100	<b>0.3</b>	0.3	257.8	0.3	<b>0.3</b>	317.8	<b>0.3</b>	0.4	336.3	0.3	<b>0.3</b>	418.6
	200	<b>0.2</b>	0.3	220.7	<b>0.3</b>	0.3	288.2	<b>0.3</b>	0.5	286.7	<b>0.3</b>	0.4	369.6
	500	<b>0.3</b>	0.5	178.7	<b>0.3</b>	0.5	252.7	<b>0.4</b>	0.7	213.7	<b>0.5</b>	0.8	317.0
	1000	<b>0.4</b>	0.7	158.1	<b>0.3</b>	0.7	222.8	<b>0.6</b>	1.1	183.3	<b>0.6</b>	1.2	268.9

Table 8. Runtime (sec.) for fixed  $p/n_0$ 

		$p/n_0 = 0.4$						$p/n_0 = 0.8$					
		$m_*/m = 0.3$			$m_*/m = 0.6$			$m_*/m = 0.3$			$m_*/m = 0.6$		
$m$	$n_0$	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL	JTT1	JTT2	NL
20	50	<b>0.1</b>	0.1	20.6	<b>0.1</b>	0.1	23.8	<b>0.1</b>	0.1	32.3	<b>0.1</b>	0.1	34.6
	100	<b>0.1</b>	0.1	22.5	<b>0.1</b>	0.1	28.1	<b>0.1</b>	0.2	42.6	<b>0.1</b>	0.2	91.8
	200	<b>0.1</b>	0.3	23.5	<b>0.1</b>	0.3	32.2	<b>0.4</b>	0.8	53.3	<b>0.4</b>	0.9	79.4
	500	<b>0.8</b>	2.4	41.7	<b>0.8</b>	2.6	55.9	<b>3.6</b>	10.2	175.4	<b>3.6</b>	10.9	309.2
	1000	<b>5.0</b>	19.1	110.1	<b>4.7</b>	18.1	174.7	<b>26.0</b>	99.6	726.1	<b>26.5</b>	90.7	1976.2
50	50	<b>0.3</b>	0.3	303.7	<b>0.2</b>	0.2	335.6	<b>0.3</b>	0.3	594.0	<b>0.3</b>	0.3	641.8
	100	<b>0.3</b>	0.4	345.1	0.3	<b>0.3</b>	399.2	<b>0.5</b>	0.7	725.8	<b>0.5</b>	0.6	1081.6
	200	<b>0.7</b>	1.0	426.5	<b>0.6</b>	0.9	504.9	<b>1.7</b>	3.0	950.6	<b>1.7</b>	2.8	1205.5
	500	<b>3.4</b>	7.2	640.2	<b>3.5</b>	8.1	828.7	<b>16.6</b>	33.2	1782.7	<b>17.1</b>	35.3	3602.5
	1000	<b>19.6</b>	54.9	1105.8	<b>20.5</b>	60.4	1883.5	<b>128.3</b>	465.5	6051.2	<b>147.0</b>	519.6	10601.3

$p \times p$  matrix, its runtime greatly increases as  $m$  or  $p$  increases. However, this is not as major an issue as with network Lasso. We have some instances of the impossible result that JTT2 is faster than JTT1, but this result is caused by numerical error from runtime being extremely short. Overall, the results of the simulation allow us to conclude that the JTT method has excellent performance compared to network Lasso in terms of clustering accuracy, estimation accuracy, and runtime.

### 5. Real data example

In this section, we compare the JTT method with network Lasso through an application to real data. Settings and implementation environment are the same as in the previous section. The dataset used is Ames Housing Data (De Cock, 2011) obtained from R package `modeldata` (ver. 1.5.1; Kuhn, 2025). We regard a variable Neighborhood, which indicates a location of housing, as a group, and define the edge set  $E$  by regional adjacency. In this application, small sample groups are merged with other groups in advance. Let the response variable be `Sale_Price` (USD), which is housing price, and the remainder be explanatory variables. Category variables among the explanatory variables are transformed to binary dummy variables with value 1 corresponding to the most frequent category. Furthermore, some variables are removed from the model to guarantee full column rank of the matrix of explanatory variables for each group. In the end, we apply the JTT method and network Lasso to Ames Housing Data with  $n = 2930$ ,  $p = 34$ ,  $m = 19$ , and  $\#(E) = 31$ , where the group-wise sample sizes are as summarized in Table 9.

Table 9. Group-wise sample sizes

Group	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Sample size	496	268	239	194	214	166	193	151	131	125	151	108	113	93	74	71	51	48	44

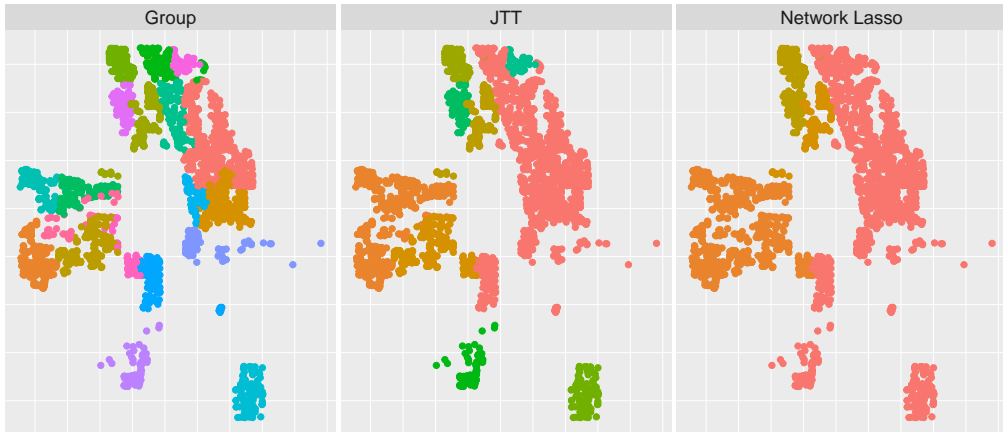


Figure 1. Initial 19 groups, and clustering results

Figure 1 shows the initial 19 groups and the clustering results for the JTT method and network Lasso, which produced 9 and 4 clusters, respectively. Table 10 summarizes the results of leave-one-

Table 10. Results of leave-one-out cross-validation

Prediction accuracy (USD)			Fitting							
			$R^2$			Clusters		Runtime (sec.)		
JTT1	JTT2	NL	JTT1	JTT2	NL	JTT	NL	JTT1	JTT2	NL
30588.0	28669.1	36500.2	0.91	0.90	0.80	8.999	3.954	0.044	0.102	104.906

out cross-validation, in which prediction accuracy is measured as  $\sqrt{\sum_{i=1}^n (y_i^* - \hat{y}_i^*)^2/n}$ , where  $y_i^*$  and  $\hat{y}_i^*$  are the  $i$ th test data and the predicted value, respectively, and other values are averages of the coefficients of determination, numbers of estimated clusters, and runtimes of fitting results for the training data. From the table, we can see that the JTT method is much faster and has higher prediction accuracy than network Lasso. Moreover, the penalized estimation method in JTT2 was able to improve prediction accuracy relative to JTT1.

## 6. Conclusion

For group-wise linear regression, we proposed the join-two-together (JTT) method for group clustering when the relationships of the groups are given by a graph. By formulating the clustering problem as the edge selection problem in the graph, we were able to formulate a simple method based on the KOO method. The performance of the JTT method depends on the model selection criterion used to evaluate the goodness of a model. This paper presented a condition such that the JTT method based on a  $GC_p$  criterion has consistency in the edge selection under a high-dimensional asymptotic framework, and proposed a specific criterion. Furthermore, to implement the JTT method, R package JTT was developed.

Simulation demonstrated that the JTT method has consistency. Furthermore, we found that a simple penalized estimation method can improve MSEs for the fitted values and the estimator of regression coefficients. Moreover, the JTT method performed much better than network Lasso in terms clustering accuracy, MSE, and runtime, in both simulation and an application to real data. Although network Lasso was implemented under the default settings of R package GGFL, its performance may be improved by changing the model selection criterion to select the tuning parameter and penalty weights. However, since we cannot expect significant improvement in calculation speed in that way, the superiority of the JTT method can be considered invariant. The present paper focused on group clustering, but selection of explanatory variables is also important. By adding a penalty for variable selection, network Lasso can be extended to perform group clustering and variable selection simultaneously. However, needless to say, its optimization becomes more complex and it requires more computation time. On the other hand, when using the JTT method to perform clustering, variable selection can be carried out using the KOO method. In that case, consistent variable selection becomes possible

with the number of calculations increasing by only  $p$  (or  $mp$ ). Furthermore, the KOO method does not require additional calculation of inverse matrices. Hence, clustering and variable selection can be performed at high speed.

However, the JTT method requires a sufficiently large sample size for each group to guarantee consistency. Similar to the Ames Housing Data utilized in Section 5, it is plausible that some small sample groups appear in practical scenarios. Therefore, we need to consider a method for coping with this problem.

**Acknowledgment** The authors thank Prof. Hirokazu Yanagihara of Osaka Metropolitan University for his many helpful comments and FORTE Science Communications (<https://www.forte-science.co.jp/>) for English language editing. This work was partially supported by JSPS KAKENHI Grant Numbers 25K17296 and 25K21159.

## References

- Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. B. N. Petrov & F. Csáki, eds, 2nd International Symposium on Information Theory. Akadémiai Kiadó. Budapest. 267–281.
- Atkinson, A. C. (1980). A note on the generalized information criterion for choice of a model. *Biometrika*, **67**, 413–418.
- Bai, Z. D., Fujikoshi, Y. & Hu, J. (2018). Strong consistency of the AIC, BIC,  $C_p$  and KOO methods in high-dimensional multivariate linear regression. Technical Report TR-No. 18-09. Hiroshima Statistical Research Group. Hiroshima.
- Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, **3**, 1–122.
- De Cock, D. (2011). Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project. *J. Stat. Educ.*, **19**.
- Fujikoshi, Y. & Satoh, K. (1997). Modified AIC and  $C_p$  in multivariate linear regression. *Biometrika*, **84**, 707–716.
- Hallac, D., Leskovec, J. & Boyd, S. (2015). Network Lasso: Clustering and optimization in large graphs. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '15. Association for Computing Machinery. New York, NY, USA. 387–396.

- Imori, S., Katayama, S. & Wakaki, H. (2014). Screening and selection methods in high-dimensional linear regression model. Technical Report TR-No. 14-01. Hiroshima Statistical Research Group. Hiroshima.
- Kuhn, M. (2025). *modeldata: Data sets useful for modeling examples*. R package version 1.5.1. **URL:** <https://CRAN.R-project.org/package=modeldata>
- Mallows, C. L. (1973). Some comments on  $C_p$ . *Technometrics*, **15**, 661–675.
- Nishii, R. (1984). Asymptotic properties of criteria for selection of variables in multiple regression. *Ann. Statist.*, **12**, 758–765.
- Nishii, R., Bai, Z. D. & Krishnaiah, P. R. (1988). Strong consistency of the information criterion for model selection in multivariate analysis. *Hiroshima Math. J.*, **18**, 451–462.
- Oda, R. & Yanagihara, H. (2020). A fast and consistent variable selection method for high-dimensional multivariate linear regression with a large number of explanatory variables. *Electron. J. Stat.*, **14**, 1386–1412.
- Oda, R. & Yanagihara, H. (2021). A consistent likelihood-based variable selection method in normal multivariate linear regression.. I. Czarnowski, R. J. Howlett & L. C. Jain, eds, Intelligent Decision Technologies. Springer Singapore. Singapore. 391–401.
- Ohishi, M. (2025a). *GGFL: Coordinate optimization for GGFL*. R package version 1.0.2. **URL:** <https://github.com/ohishim/GGFL>
- Ohishi, M. (2025b). *JTT: Consistent clustering for group-wise linear regression*. R package version 0.1.0. **URL:** <https://github.com/ohishim/JTT>
- Ohishi, M., Fukui, K., Okamura, K., Itoh, Y. & Yanagihara, H. (2021). Coordinate optimization for generalized fused Lasso. *Comm. Statist. Theory Methods*, **50**, 5955–5973.
- Ohishi, M., Okamura, K., Itoh, Y., Wakaki, H. & Yanagihara, H. (2025). Coordinate descent algorithm for generalized group fused Lasso. *Behaviormetrika*, **52**, 105–137.
- Ohishi, M., Yanagihara, H. & Fujikoshi, Y. (2020). A fast algorithm for optimizing ridge parameters in a generalized ridge regression by minimizing a model selection criterion. *J. Statist. Plann. Inference*, **204**, 187–205.
- R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. **URL:** <https://www.R-project.org/>
- Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, **6**, 461–464.

- Wang, H. & Leng, C. (2008). A note on adaptive group Lasso. *Comput. Statist. Data Anal.*, **52**, 5277–5286.
- Yanagihara, H. (2016). A high-dimensionality-adjusted consistent  $C_p$ -type statistic for selecting variables in a normality-assumed linear regression with multiple responses. *Procedia Comput. Sci.*, **96**, 1096–1105.
- Yanagihara, H. (2018). Explicit solution to the minimization problem of generalized cross-validation criterion for selecting ridge parameters in generalized ridge regression. *Hiroshima Math. J.*, **48**, 203–222.
- Yanagihara, H. (2024). High-dimensionality-adjusted consistent AIC in normal multivariate linear regression. *Procedia Comput. Sci.*, **246**, 2022–2031.
- Zhao, L. C., Krishnaiah, P. R. & Bai, Z. D. (1986). On detection of the number of signals in presence of white noise. *J. Multivariate Anal.*, **20**, 1–25.

## Appendix

### A.1. Proof of equation (2.4)

Let  $\mathbf{A}_{k\ell} = \mathbf{M}_k \mathbf{M}_{k\ell}^{-1} \mathbf{M}_\ell$ . Since  $\mathbf{M}_{k\ell} = \mathbf{M}_k + \mathbf{M}_\ell$ , using the identity  $\mathbf{A} - \mathbf{A}(\mathbf{A} + \mathbf{B})^{-1} \mathbf{A} = \mathbf{A}(\mathbf{A} + \mathbf{B})^{-1} \mathbf{B}$  for nonsingular matrices  $\mathbf{A}$  and  $\mathbf{B}$ , the coefficient matrix of the quadratic form  $\delta_{k\ell}$  ( $(k, \ell) \notin E_*$ ) for  $(\beta_k^{*'}, \beta_\ell^{*'})'$  can be expressed as

$$\begin{aligned} & \begin{pmatrix} \mathbf{X}'_k & \mathbf{O}_{p, n_\ell} \\ \mathbf{O}_{p, n_k} & \mathbf{X}'_\ell \end{pmatrix} (\mathbf{I}_{n_{k\ell}} - \mathbf{P}_{k\ell}) \begin{pmatrix} \mathbf{X}_k & \mathbf{O}_{n_k, p} \\ \mathbf{O}_{n_\ell, p} & \mathbf{X}_\ell \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{M}_k - \mathbf{M}_k \mathbf{M}_{k\ell}^{-1} \mathbf{M}_k & -\mathbf{M}_k \mathbf{M}_{k\ell}^{-1} \mathbf{M}_\ell \\ -\mathbf{M}_\ell \mathbf{M}_{k\ell}^{-1} \mathbf{M}_k & \mathbf{M}_\ell - \mathbf{M}_\ell \mathbf{M}_{k\ell}^{-1} \mathbf{M}_\ell \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{k\ell} & -\mathbf{A}_{k\ell} \\ -\mathbf{A}_{k\ell} & \mathbf{A}_{k\ell} \end{pmatrix}, \end{aligned}$$

where  $\mathbf{O}_{n,p}$  is an  $n \times p$  matrix of zeros. Since  $\mathbf{A}_{k\ell}$  is a symmetric matrix, for  $\gamma_{k\ell} = \beta_k^* - \beta_\ell^*$ , we have

$$\begin{aligned} \sigma_*^2 \delta_{k\ell} &= (\beta_k^{*'}, \beta_\ell^{*'}) \begin{pmatrix} \mathbf{A}_{k\ell} & -\mathbf{A}_{k\ell} \\ -\mathbf{A}_{k\ell} & \mathbf{A}_{k\ell} \end{pmatrix} \begin{pmatrix} \beta_k^* \\ \beta_\ell^* \end{pmatrix} = \beta_k^{*'} \mathbf{A}_{k\ell} \beta_k^* - 2\beta_k^{*'} \mathbf{A}_{k\ell} \beta_\ell^* + \beta_\ell^{*'} \mathbf{A}_{k\ell} \beta_\ell^* = \gamma'_{k\ell} \mathbf{A}_{k\ell} \gamma_{k\ell} \\ &\geq \lambda_{\min}(\mathbf{A}_{k\ell}) \|\gamma_{k\ell}\|^2. \end{aligned}$$

Hence, Assumptions A2 and A3 imply

$$\begin{aligned} n_0^{-c_3} \delta_{\min} &\geq n_0^{-c_3} \min_{(k, \ell) \notin E_*} \lambda_{\min}(\mathbf{A}_{k\ell}) \|\gamma_{k\ell}\|^2 / \sigma_*^2 \\ &\geq \left\{ n_0^{-1} \min_{(k, \ell) \notin E_*} \lambda_{\min}(\mathbf{A}_{k\ell}) \right\} \left\{ n_0^{1-c_3} \min_{(k, \ell) \notin E_*} \|\gamma_{k\ell}\|^2 / \sigma_*^2 \right\} \geq c_1 c_2, \end{aligned}$$

and (2.4) is proved.

## A.2. Proof of Theorem 1

We prove the theorem with the following lemma given in Oda & Yanagihara (2020).

**Lemma A.1.** *Let  $u_1$ ,  $u_2$ , and  $v$  be random variables distributed according to  $\chi^2(p)$ ,  $\chi^2(p, \delta)$ , and  $\chi^2(N)$ , respectively, where  $u_1$  and  $u_2$  are independent of  $v$  and  $N = n - mp$ . Then, for  $N - 4r > 0$  ( $r \in \mathbb{N}$ ), the following expressions are true for  $N \rightarrow \infty$ .*

$$\begin{aligned} \mathbb{E} \left[ \left( \frac{u_1}{v} - \frac{p}{N-2} \right)^{2r} \right] &= O(p^r n^{-2r}), \\ \mathbb{E} \left[ \left( \frac{u_2}{v} - \frac{p+\delta}{N-2} \right)^{2r} \right] &= O \left( \max \left\{ (p+\delta)^r n^{-2r}, (p+\delta)^{2r} n^{-3r} \right\} \right). \end{aligned}$$

From (2.2), the probability of  $\hat{E}(\alpha) = E_*$  can be evaluated as follows:

$$\begin{aligned} \mathbb{P}(\hat{E}(\alpha) = E_*) &= \mathbb{P} \left( \left\{ \bigcap_{(k,\ell) \in E_*} \{GC_p^{(k\ell)}(\alpha) \leq GC_p^{(0)}(\alpha)\} \right\} \cap \left\{ \bigcap_{(k,\ell) \notin E_*} \{GC_p^{(k\ell)}(\alpha) > GC_p^{(0)}(\alpha)\} \right\} \right) \\ &\geq 1 - \sum_{(k,\ell) \in E_*} \mathbb{P}(GC_p^{(k\ell)}(\alpha) > GC_p^{(0)}(\alpha)) - \sum_{(k,\ell) \notin E_*} \mathbb{P}(GC_p^{(k\ell)}(\alpha) < GC_p^{(0)}(\alpha)). \end{aligned}$$

Hence, under the asymptotic framework in (2.3), it is sufficient to show that

$$\sum_{(k,\ell) \in E_*} \mathbb{P}(GC_p^{(k\ell)}(\alpha) > GC_p^{(0)}(\alpha)) = o(1), \quad \sum_{(k,\ell) \notin E_*} \mathbb{P}(GC_p^{(k\ell)}(\alpha) < GC_p^{(0)}(\alpha)) = o(1). \quad (\text{A.1})$$

The difference of  $GC_p$  criteria in (2.1) is given by

$$\begin{aligned} GC_p^{(k\ell)}(\alpha) - GC_p^{(0)}(\alpha) &= \frac{1}{s^2} \left\{ \mathbf{y}'_{k\ell} (\mathbf{I}_{n_{k\ell}} - \mathbf{P}_{k\ell}) \mathbf{y}_{k\ell} - \sum_{j \notin V_{k\ell}} \mathbf{y}'_j (\mathbf{I}_{n_j} - \mathbf{P}_j) \mathbf{y}_j \right\} - \alpha p \\ &= N \frac{\mathbf{y}'_{k\ell} \{\text{diag}(\mathbf{P}_k, \mathbf{P}_\ell) - \mathbf{P}_{k\ell}\} \mathbf{y}_{k\ell}}{\sum_{j=1}^m \mathbf{y}'_j (\mathbf{I}_{n_j} - \mathbf{P}_j) \mathbf{y}_j} - \alpha p, \end{aligned}$$

where  $\text{diag}(\mathbf{A}, \mathbf{B})$  indicates the block diagonal matrix with  $\mathbf{A}$  and  $\mathbf{B}$  as the diagonal blocks. Let  $u_{k\ell} = \mathbf{y}'_{k\ell} \{\text{diag}(\mathbf{P}_k, \mathbf{P}_\ell) - \mathbf{P}_{k\ell}\} \mathbf{y}_{k\ell} / \sigma_*^2$  and  $v = \sum_{j=1}^m \mathbf{y}'_j (\mathbf{I}_{n_j} - \mathbf{P}_j) \mathbf{y}_j / \sigma_*^2$ . Then  $u_{k\ell}$  and  $v$  can be rewritten as quadratic forms for  $\mathbf{y} = (\mathbf{y}'_1, \dots, \mathbf{y}'_m)'$ . Their coefficient matrices are idempotent and their product is  $\mathbf{O}_{n,n}$ . Hence, Cochran's theorem gives that  $u_{k\ell}$  and  $v$  are independent and that  $u_{k\ell} \sim \chi^2(p, \delta_{k\ell})$  and  $v \sim \chi^2(N)$ . Recall that  $\delta_{k\ell} = 0$  holds when  $(k, \ell) \in E_*$ . Hence, we have

$$GC_p^{(k\ell)}(\alpha) - GC_p^{(0)}(\alpha) = \begin{cases} \frac{u_0 N}{v} - \alpha p & ((k, \ell) \in E_*) \\ \frac{u_{k\ell} N}{v} - \alpha p & ((k, \ell) \notin E_*) \end{cases}, \quad (\text{A.2})$$

where  $u_0$  is independent of  $v$  and  $u_0 \sim \chi^2(p)$ .

We first show (A.1) when  $(k, \ell) \in E_*$ . Regarding the first equation in (A.1), we have

$$\begin{aligned} \sum_{(k, \ell) \in E_*} \mathbb{P}\left(GC_p^{(k\ell)}(\alpha) > GC_p^{(0)}(\alpha)\right) &= q_* \mathbb{P}\left(\frac{u_0}{v} > \frac{\alpha p}{N}\right) = q_* \mathbb{P}\left(\frac{u_0}{v} - \frac{p}{N-2} > \rho\right) \\ &\leq q_* \mathbb{P}\left(\left|\frac{u_0}{v} - \frac{p}{N-2}\right| \geq \rho\right), \end{aligned}$$

where  $q_* = \#(E_*)$ ,  $\rho = \beta p/N$ , and  $\beta = \alpha - N/(N-2)$ . For  $r_1 \in \mathbb{N}$ , it holds from Markov's inequality and Lemma A.1 that

$$\mathbb{P}\left(\left|\frac{u_0}{v} - \frac{p}{N-2}\right| \geq \rho\right) \leq \rho^{-2r_1} \mathbb{E}\left[\left(\frac{u_0}{v} - \frac{p}{N-2}\right)^{2r_1}\right] = O(\beta^{-2r_1} p^{-r_1}).$$

Since  $q_* = O(m^2)$  and  $\beta p^{1/2}/m^{1/r_1} \rightarrow \infty$ , we have

$$\sum_{(k, \ell) \in E_*} \mathbb{P}\left(GC_p^{(k\ell)}(\alpha) > GC_p^{(0)}(\alpha)\right) = O(m^2 \beta^{-2r_1} p^{-r_1}) = o(1). \quad (\text{A.3})$$

Regarding the second equation in (A.1), when  $(k, \ell) \notin E_*$ , we have

$$\sum_{(k, \ell) \notin E_*} \mathbb{P}\left(GC_p^{(k\ell)}(\alpha) < GC_p^{(0)}(\alpha)\right) = \sum_{(k, \ell) \notin E_*} \mathbb{P}\left(\frac{u_{k\ell}}{v} < \frac{\alpha p}{N}\right) = \sum_{(k, \ell) \notin E_*} \mathbb{P}\left(\frac{u_{k\ell}}{v} - \frac{p + \delta_{k\ell}}{N-2} < \rho - \frac{\delta_{k\ell}}{N-2}\right).$$

Since (2.4) and  $\beta p/n_0^{c_3} \rightarrow 0$  imply  $\rho - \delta_{k\ell}/(N-2) < 0$  for sufficiently large  $N$ , we have

$$\mathbb{P}\left(\frac{u_{k\ell}}{v} - \frac{p + \delta_{k\ell}}{N-2} < \rho - \frac{\delta_{k\ell}}{N-2}\right) \leq \mathbb{P}\left(\left|\frac{u_{k\ell}}{v} - \frac{p + \delta_{k\ell}}{N-2}\right| \geq \frac{\delta_{k\ell}}{N-2} - \rho\right).$$

Furthermore, for  $r_2 \in \mathbb{N}$ , it holds from Markov's inequality and Lemma A.1 that

$$\begin{aligned} \sum_{(k, \ell) \notin E_*} \mathbb{P}\left(\left|\frac{u_{k\ell}}{v} - \frac{p + \delta_{k\ell}}{N-2}\right| \geq \frac{\delta_{k\ell}}{N-2} - \rho\right) &\leq \sum_{(k, \ell) \notin E_*} \left(\frac{\delta_{k\ell}}{N-2} - \rho\right)^{-2r_2} \mathbb{E}\left[\left(\frac{u_{k\ell}}{v} - \frac{p + \delta_{k\ell}}{N-2}\right)^{2r_2}\right] \\ &\leq (q - q_*) \max_{(k, \ell) \notin E_*} \left(\frac{\delta_{k\ell}}{N-2} - \rho\right)^{-2r_2} \mathbb{E}\left[\left(\frac{u_{k\ell}}{v} - \frac{p + \delta_{k\ell}}{N-2}\right)^{2r_2}\right], \\ \mathbb{E}\left[\left(\frac{u_{k\ell}}{v} - \frac{p + \delta_{k\ell}}{N-2}\right)^{2r_2}\right] &= O\left(\max\left\{\frac{(p + \delta_{k\ell})^{r_2}}{n^{2r_2}}, \frac{(p + \delta_{k\ell})^{2r_2}}{n^{3r_2}}\right\}\right). \end{aligned}$$

Notice that the following inequalities hold for sufficiently large  $N$ :

$$\left(\frac{N-2}{\delta_{k\ell}}\right)^{2r_2} \left(\frac{p + \delta_{k\ell}}{n^2}\right)^{r_2} \leq \left(\frac{1 + p/\delta_{\min}}{\delta_{\min}}\right)^{r_2}, \quad \left(\frac{N-2}{\delta_{k\ell}}\right)^{2r_2} \left(\frac{p + \delta_{k\ell}}{n^{3/2}}\right)^{2r_2} \leq \left(\frac{1 + p/\delta_{\min}}{n^{1/2}}\right)^{2r_2}.$$

Hence, we have

$$\begin{aligned} &(q - q_*) \max_{(k, \ell) \notin E_*} \left(\frac{\delta_{k\ell}}{N-2} - \rho\right)^{-2r_2} \left\{ \frac{(p + \delta_{k\ell})^{r_2}}{n^{2r_2}} + \frac{(p + \delta_{k\ell})^{2r_2}}{n^{3r_2}} \right\} \\ &\leq (q - q_*) \left(1 - \frac{\rho(N-2)}{\delta_{\min}}\right)^{-2r_2} \left\{ \left(\frac{1 + p/\delta_{\min}}{\delta_{\min}}\right)^{r_2} + \left(\frac{1 + p/\delta_{\min}}{n^{1/2}}\right)^{2r_2} \right\} \\ &= O\left(m^2 \delta_{\min}^{-r_2}\right) + O\left(m^2 p^{r_2} \delta_{\min}^{-2r'}\right) + O\left(m^2 n^{-r_2}\right) + O\left(m^2 p^{2r_2} n^{-r_2} \delta_{\min}^{-2r'}\right) \\ &= O\left(n_0^{2-c_3 r_2} + n_0^{2+r_2-2c_3 r_2} + m^{2-r_2} n_0^{-r_2} + m^{2-r_2} n_0^{r_2-2c_3 r_2}\right). \end{aligned} \quad (\text{A.4})$$

If  $r_2 > 2$  and  $c_3 > (r_2 + 2)/2r_2$ , then (A.4) converges to 0. Since  $r_2$  is arbitrary and  $(r_2 + 2)/2r_2 > 1/2$  holds for  $r_2 > 2$ , it follows that  $c_3 > (r_2 + 2)/2r_2$  is equivalent to  $c_3 \geq 1/2$ . Hence, we have

$$\sum_{(k,\ell) \notin E_*} \mathbb{P}\left(GC_p^{(k\ell)}(\alpha) < GC_p^{(0)}(\alpha)\right) = o(1).$$

As discussed above, (A.1) holds and the probability of  $\hat{E}(\alpha) = E_*$  converges to 1. Moreover, we have the convergence order from (A.3) and (A.4), and consequently Theorem 1 is proved.

### A.3. Details of $B$ in (3.2)

Based on Yanagihara (2024), we explain how  $B$ , which is included in  $\hat{\beta}$  in (3.2), standardizes the differences of  $HCGC_p$  criteria. Based on (A.2), we define  $D_{k\ell}(\alpha)$  as

$$D_{k\ell}(\alpha) = \frac{u_{k\ell}}{v} = \frac{1}{N} \left\{ HCGC_p^{(k\ell)}(\alpha) - HCGC_p^{(0)}(\alpha) + \alpha p \right\}.$$

Notice that  $\mathbb{E}[D_{k\ell}(\hat{\alpha})] = \mu = p/(N - 2)$  and  $\text{Var}[D_{k\ell}(\hat{\alpha})] = \tau^2 = 2pB^2/N^2$  from  $u_{k\ell} \sim \chi^2(p)$  and  $v \sim \chi^2(N)$  when  $(k, \ell) \in E_*$ . Hence, we have  $N^{-1}\{HCGC_p^{(k\ell)}(\hat{\alpha}) - HCGC_p^{(0)}(\hat{\alpha})\} = D_{k\ell}(\hat{\alpha}) - \mu - \tau m^{1/4} \log n_0 / \sqrt{2}$ . This expression implies

$$\mathbb{P}\left(HCGC_p^{(k\ell)}(\hat{\alpha}) - HCGC_p^{(0)}(\hat{\alpha}) > 0\right) = \mathbb{P}\left(\frac{D_{k\ell}(\hat{\alpha}) - \mu}{\tau} > \frac{m^{1/4} \log n_0}{\sqrt{2}}\right),$$

and we can see that  $B$  standardizes the difference of  $HCGC_p$  criteria with  $\alpha = \hat{\alpha}$ .

### A.4. Proof of Theorem 2

We use the following lemma to prove the theorem (e.g., Imori *et al.*, 2014).

**Lemma A.2.** *Let  $z$ ,  $u_1$ , and  $u_2$  be random variables distributed according to  $N(0, 1)$ ,  $\chi^2(p)$ , and  $\chi^2(p, \delta)$ , respectively. Then, we have*

$$\begin{aligned} \mathbb{P}(|z| > r) &\leq \exp(-r^2/2), \\ \mathbb{P}(u_1 > (1+r)p) &\leq \exp(-hrp) \quad (r \in [1, \infty), h = (1 - \log 2)/2), \\ \mathbb{P}(u_1 < (1-r)p) &\leq \exp(-rp/4) \quad (r \in [0, 1]), \\ \mathbb{P}(u_2 < r) &\leq \mathbb{P}(|z| > t) + \mathbb{P}(u_1 < r - \delta + 2\sqrt{\delta}t) \quad (t \geq 0). \end{aligned}$$

Similar to the proof of Theorem 1, it is sufficient to show (A.1). We first evaluate the probability for  $(k, \ell) \in E_*$  for  $r_1 \in (0, 1)$  as

$$\begin{aligned} \sum_{(k,\ell) \in E_*} \mathbb{P}\left(GC_p^{(k\ell)}(\alpha) > GC_p^{(0)}(\alpha)\right) &= q_* \mathbb{P}\left(\frac{u_0}{v} > \frac{(1-r_1)\alpha p}{(1-r_1)N}\right) \\ &\leq q_* \mathbb{P}(u_0 > (1-r_1)\alpha p) + q_* \mathbb{P}(v < (1-r_1)N). \end{aligned} \quad (\text{A.5})$$

Regarding the first term in (A.5), since  $(1 - r_1)\alpha = 1 + \{(1 - r_1)\alpha - 1\}$ , it holds from Lemma A.2 that

$$q_* \mathbb{P}(u_0 > (1 - r_1)\alpha p) \leq q_* \exp[-h\{(1 - r_1)\alpha - 1\}p] = O(m^2 \exp[-h\{(1 - r_1)\alpha - 1\}p]),$$

for  $\alpha \geq 2/(1 - r_1)$ . Hence, we have

$$m^2 \exp[-h\{(1 - r_1)\alpha - 1\}p] = \exp\left(-\alpha p \left[-\frac{2 \log m}{\alpha p} + h \left\{(1 - r_1) - \frac{1}{\alpha}\right\}\right]\right) = o(1),$$

from  $(1 - r_1) - 1/\alpha > 0$  and  $\alpha p / \log m \rightarrow \infty$ . On the other hand, regarding the second term in (A.5), it holds from Lemma A.2 that

$$q_* \mathbb{P}(v < (1 - r_1)N) \leq q_* \exp(-r_1 N/4) = O(m^2 \exp(-r_1 N/4)) = o(1).$$

Hence, we have

$$\sum_{(k, \ell) \in E_*} \mathbb{P}(GC_p^{(k\ell)}(\alpha) > GC_p^{(0)}(\alpha)) = o(1).$$

Looking at  $(k, \ell) \notin E_*$ , for  $r_2 \in [1, \infty)$ , we have

$$\sum_{(k, \ell) \notin E_*} \mathbb{P}(GC_p^{(k\ell)}(\alpha) < GC_p^{(0)}(\alpha)) \leq \sum_{(k, \ell) \notin E_*} \mathbb{P}(u_{k\ell} < (1 + r_2)\alpha p) + (q - q_*) \mathbb{P}(v > (1 + r_2)N). \quad (\text{A.6})$$

Regarding the first term of the RHS in the above expression, for  $z \sim N(0, 1)$  and  $t (\geq 0)$ , it holds from Lemma A.2 that

$$\mathbb{P}(u_{k\ell} < (1 + r_2)\alpha p) \leq \mathbb{P}(|z| > t) + \mathbb{P}\left(u_0 \leq (1 + r_2)\alpha p - \delta_{k\ell} + 2\sqrt{\delta_{k\ell}t}\right).$$

Since  $\alpha p / n_0^{c_3} \rightarrow 0$ , (2.4) and Lemma A.2, for  $t = \{c_1 c_2 n_0^{c_3} - (1 + r_2)\alpha p\} / 2\sqrt{c_1 c_2 n_0^{c_3}}$  and sufficiently large  $n_0$ , we have

$$\begin{aligned} \sum_{(k, \ell) \notin E_*} \mathbb{P}(u_{k\ell} < (1 + r_2)\alpha p) &\leq \sum_{(k, \ell) \notin E_*} \mathbb{P}(|z| > t) \leq \sum_{(k, \ell) \notin E_*} \exp(-t^2) \\ &= (q - q_*) \exp\left[-\frac{c_1 c_2 n_0^{c_3}}{8} \left\{1 - \frac{(1 + r_2)\alpha p}{c_1 c_2 n_0^{c_3}}\right\}^2\right] \\ &= O(m^2 \exp(-c_1 c_2 n_0^{c_3} / 8)) = o(1). \end{aligned}$$

On the other hand, regarding the second term of the RHS in (A.6), it holds from Lemma A.2 that

$$(q - q_*) \mathbb{P}(v > (1 + r_2)N) \leq (q - q_*) \exp(-hr_2 N) = O(m^2 \exp(-hr_2 N)) = o(1).$$

Hence, we have

$$\sum_{(k, \ell) \notin E_*} \mathbb{P}(GC_p^{(k\ell)}(\alpha) < GC_p^{(0)}(\alpha)) = o(1).$$

Consequently, (A.1) holds and Theorem 2 is proved.

### A.5. Proof of Theorem 3

We define  $(k_1, \ell_1) \in E_*$  and  $(k_2, \ell_2) \notin E_*$  such that, respectively,

$$\begin{aligned} (k_1, \ell_1) \in E_* \text{ s.t. } & \mathbb{P}\left(GC_p^{(k_1 \ell_1)}(\alpha) < GC_p^{(0)}(\alpha)\right) \not\rightarrow 1, \\ (k_2, \ell_2) \notin E_* \text{ s.t. } & \mathbb{P}\left(GC_p^{(k_2 \ell_2)}(\alpha) > GC_p^{(0)}(\alpha)\right) \not\rightarrow 1. \end{aligned}$$

Then, we have

$$\begin{aligned} \mathbb{P}\left(\hat{E}(\alpha) = E_*\right) &= \mathbb{P}\left\{\left\{\bigcap_{(k, \ell) \in E_*} \left\{GC_p^{(k \ell)}(\alpha) \leq GC_p^{(0)}(\alpha)\right\}\right\} \cap \left\{\bigcap_{(k, \ell) \notin E_*} \left\{GC_p^{(k \ell)}(\alpha) > GC_p^{(0)}(\alpha)\right\}\right\}\right\} \\ &\leq \left\{\begin{array}{l} \mathbb{P}\left(GC_p^{(k_1 \ell_1)}(\alpha) < GC_p^{(0)}(\alpha)\right) \\ \mathbb{P}\left(GC_p^{(k_2 \ell_2)}(\alpha) > GC_p^{(0)}(\alpha)\right) \end{array}\right\} \not\rightarrow 1. \end{aligned}$$

Hence, it is sufficient to show that either  $(k_1, \ell_1)$  or  $(k_2, \ell_2)$  exists.

From (A.2), for all  $(k, \ell) \in E_*$ , we have

$$\mathbb{P}\left(GC_p^{(k \ell)}(\alpha) < GC_p^{(0)}(\alpha)\right) = \mathbb{P}\left(\frac{u_0}{v} < \frac{\alpha p}{N}\right).$$

Notice that  $v/N$  converges to 1 in probability, and  $(u_0/p)/(v/N)$  also converges to 1 in probability as  $p \rightarrow \infty$ . Hence, we have

$$\left\{\begin{array}{l} p: \text{ fixed and } \alpha \rightarrow \infty \implies \mathbb{P}\left(\frac{u_0}{v} < \frac{\alpha p}{N}\right) = \mathbb{P}\left(\frac{u_0}{v/N} < \alpha p\right) \not\rightarrow 1 \\ p \rightarrow \infty \text{ and } \liminf \alpha < 1 \implies \mathbb{P}\left(\frac{u_0}{v} < \frac{\alpha p}{N}\right) = \mathbb{P}\left(\frac{u_0/p}{v/N} < \alpha\right) \not\rightarrow 1 \end{array}\right.$$

Let  $(k_2, \ell_2) = \arg \min_{(k, \ell) \notin E_*} \delta_{k \ell}$ . Then we have

$$\mathbb{P}\left(GC_p^{(k_2 \ell_2)}(\alpha) > GC_p^{(0)}(\alpha)\right) = \mathbb{P}\left(\frac{u_{k_2 \ell_2}}{v} > \frac{\alpha p}{N}\right).$$

When  $\delta_{\min}/p \rightarrow \infty$ , it holds from Markov's inequality and  $\alpha p/\delta_{\min} \rightarrow \infty$  that

$$\mathbb{P}\left(\frac{u_{k_2 \ell_2}}{v} > \frac{\alpha p}{N}\right) \leq \frac{N}{\alpha p} \cdot \frac{p + \delta_{\min}}{N - 2} = O(\delta_{\min}/\alpha p) = o(1).$$

On the other hand, when  $\delta_{\min}/p \rightarrow c_4$  ( $\in [0, \infty)$ ), since  $\mathbb{E}[u_{k_2 \ell_2}/p] = 1 + \delta_{\min}/p \rightarrow 1 + c_4$  and  $\text{Var}[u_{k_2 \ell_2}/p] = 2/p + 4\delta_{\min}/p^2 \rightarrow 0$ ,  $u_{k_2 \ell_2}/p$  converges to  $1 + c_4$  in probability. Hence, since  $v/N$  converges to 1 in probability and  $\limsup \alpha > 1 + c_4$ , we have

$$\mathbb{P}\left(\frac{u_{k_2 \ell_2}}{v} > \frac{\alpha p}{N}\right) = \mathbb{P}\left(\frac{u_{k_2 \ell_2}/p}{v/N} > \alpha\right) \not\rightarrow 1.$$

Consequently, Theorem 3 is proved.