mathematics

MDPI

Article

Fast Cluster Bootstrap Methods for Spatial Error Models

Yu Zheng [1] and Honggang Fan [2,*]

[1] School of Science, China University of Geosciences (Beijing), Beijing 100083, China; zhengyu@cugb.edu.cn
[2] School of Mathematics, Renmin University of China, Beijing 100872, China
* Correspondence: fanhg2@ruc.edu.cn

Abstract

Typically, the traditional bootstrap methods for parameter inference of spatial error models suffer from high computational costs, so this study proposes fast cluster bootstrap methods for spatial error models to deal with the dilemma. The key idea is to calculate the sufficient statistics for each cluster before performing the bootstrap loop of the spatial error model, and based on these sufficient statistics, all quantities needed for bootstrap inference can be computed. Furthermore, this study performed Monte Carlo simulations, and the result reveals that compared with traditional bootstrap methods, our proposed methods can reduce the computational cost substantially and improve the reliability for obtaining the bootstrap test statistics and confidence intervals of the parameters for spatial error models.

Keywords: spatial error model; pairs cluster bootstrap; wild cluster bootstrap; computational cost

MSC: 62F40; 62J05

1. Introduction

Traditionally, maximum likelihood estimation is typically used to perform parameter inference for spatial regression models. Lim et al. [1] proposed the maximum block independent likelihood estimator based on independent sub-blocks in spatial models. As for the spatial autoregressive model, Wang and Song [2] developed a quasi-maximum likelihood method with a penalty to achieve parameter estimation in the presence of missing responses. Al-Momani and Arashi [3] constructed the maximum likelihood estimator for the parameters of the spatial error model. However, the computational cost of parameter inference for spatial regression models based on maximum likelihood estimation is relatively high, especially for high-dimensional data. Furthermore, spatial dependencies generally increase the complexity of the computation.

Typically, the objects or observations collected from the same geographic region are more comparable than those collected from distant regions, which can be regarded as a cluster, and the objects or observations within the same cluster are more comparable to each other than to those in other clusters. Spatial regression models, incorporating various spatial dependencies, provide an opportunity to analyze the objects or observations collected from the same geographic location, known as a site [3], which is similar to the idea of clusters described above.

Moreover, inference based on cluster-robust variance estimators (CRVEs) performs well when large-sample theory serves as a good guide to the finite-sample properties of the CRVE. Nevertheless, when the number of clusters is small, the size of the clusters and the characteristics of the regressors and regressand are not fairly homogeneous, the

check for updates

Academic Editor: Jin-Ting Zhang

Received: 17 July 2025
Revised: 25 August 2025
Accepted: 6 September 2025
Published: 9 September 2025

Citation: Zheng, Y.; Fan, H. Fast Cluster Bootstrap Methods for Spatial Error Models. *Mathematics* **2025**, *13*, 2913. https://doi.org/10.3390/math13182913

CRVE becomes seriously unreliable, and there is a great deal of theoretical evidence and simulation results in the latest papers about this problem [4–9].

In fact, one of the alternative methods for spatial regression models is the bootstrap method, which involves generating a number of bootstrap samples mimicking the actual sample distribution, and the bootstrap test statistics and confidence intervals for each sample are acquired with the same test procedure as for the original sample [10–14]. There are two bootstrap methods based on clustered data, including *pairs cluster bootstrap* and the *wild cluster bootstrap*, which were originally proposed by Freedman [15], Liu [16], and Mammen [17], respectively, and further extended by Davidson and Flachaire [18] and Cameron et al. [19] and have been proved to be asymptotically valid [8,20]. Furthermore, these bootstrap methods are widely applied to various problems. Bouzebda et al. [21] extended the existing theory on the bootstrap of the M-estimators, and they proposed an exchangeably weighted bootstrap for function-valued estimators defined as a zero point of a function-valued random criterion function subsequently [22].

However, applying these traditional bootstrap methods to perform bootstrapping on linear regression models for clustered data still suffers from high computational costs. MacKinnon [23] has proposed an efficient computational algorithm for bootstrapping linear regression models based on clustered data. Inspired by MacKinnon [23], this study uses a similar idea to deal with the high computational costs of parameter inference for spatial regression models.

In addition, one of the most widely used spatial regression models is the spatial error model (SEM), which models the mean of the spatial response variable by using a linear regression with a spatially lagged autoregressive error component [24].

Considering the notable gap in the literature, this study proposes fast cluster bootstrap methods for SEM to reduce the computational cost and improve the reliability for obtaining the bootstrap test statistics and confidence intervals of SEM. Specifically, before performing the bootstrap loop of SEM, the sufficient statistics are computed for each of the clusters, and based on these sufficient statistics, it is possible to calculate all of the quantities needed for bootstrap inference. In addition, the test statistics and the bootstrap confidence intervals are associated with the samples solely by these sufficient statistics.

The key contributions of this study are presented as follows:

Firstly, this is the first study to propose fast cluster bootstrap methods to overcome the high computational cost of parameter inference for SEM. Secondly, this study simulates the computational cost, two-sided equal-tailed coverage frequencies, and the precision of the bootstrap confidence intervals of the parameter with various bootstrap methods, respectively. The result reveals that the computational cost of our proposed methods is substantially reduced, and the optimal coverage of the parameter is higher than the nominal frequency. Finally, compared with traditional methods, the methods proposed in this study are conceptually simple, easier to understand, and can also be used in other spatial models.

The remainder of this paper is organized as follows. In Section 2, we present an overview of the SEM. In Section 3, we describe bootstrap computations for maximum likelihood estimation of SEM, including the fast *pairs cluster bootstrap* method for SEM and the fast unrestricted and restricted wild cluster bootstrap methods for SEM, respectively. In Section 4, we perform simulation experiments and show that using these methods can yield substantial computational savings. In Section 5, we investigate the finite-sample performance of the two-sided equal-tailed coverage frequencies and average widths of the bootstrap confidence intervals of the parameter for SEM based on various bootstrap methods. In Section 6, we give the conclusions. Finally, in Section 7, we present the future work of this study. In addition, in order to improve clarity, we provide a summary of key notations in Table 1.

Table 1. Summary of key notations.

Notation	Description	Notation	Description
L	Number of clusters	$s(\gamma)_l$	Score vector of the l^{th} cluster
γ	Spatial dependence parameter	$\hat{s}(\gamma)_l$	MLE of $s(\gamma)_l$
W	Spatial weight matrix	$\hat{s}(\gamma)_l^{*j}$	Bootstrap version of $s(\gamma)_l$
n	Sample size	β	Vector of unknown regression parameters
B	Bootstrap replications	β_0	True value of β
n_l	Number of observations for the l^{th} cluster	$\hat{\beta}$	MLE of β
X_l	Matrix of exogenous regressors	$\tilde{\beta}$	Restricted MLE of β
y_l	Vector of observations collected for the l^{th} cluster	$\hat{\beta}(\gamma)^{*j}$	Bootstrap version of $\hat{\beta}$
u_l	Vector of spatially autocorrelated remainder error for the l^{th} cluster	t^{*j}	Bootstrap t-test statistic
ε_l	Vector of the noise for the l^{th} cluster		

2. Spatial Error Model

Let $\mathcal{L} = \{1, 2, \ldots\ldots, L\}$ represent a set of L disjoint clusters, which are often referred to as locations, regions, etc. The l^{th} cluster is assumed to have n_l observations, and the sample size is $n = \sum_{l=1}^{L} n_l$. The SEM can be written as

$$y_l = X_l\beta + u_l$$

with

$$u_l = \gamma W u_l + \varepsilon_l, \quad l = 1, 2, \ldots\ldots, L, \tag{1}$$

where X_l is an $n_l \times r$ matrix of exogenous regressors, β represents a $r \times 1$ vector of unknown regression parameters, y_l is an $n_l \times 1$ vector of observations collected for the l^{th} cluster, and u_l denotes an $n_l \times 1$ vector of spatially autocorrelated remainder error. The error term involves the spatial dependence parameter γ, the $n_l \times n_l$ matrix of spatial weight matrix W, and an $n_l \times 1$ vector of the noise ε_l.

Furthermore, we make the following assumptions:

Assumption 1. *the noise ε_l has a normal distribution $N(0, \sigma^2 I)$, where I is the $n_l \times n_l$ identity matrix.*

Assumption 2. *the elements of the main diagonal of spatial weight matrix W are zeros, and if position j is adjacent to position i, the off-diagonal element $w_{ij} = 1$; otherwise, $w_{ij} = 0$ for $i \neq j$, and the normalization of W is row normalized as $W = \{\frac{w_{ij}}{\sum_{j=1}^{n_l} w_{ij}}\}_{i=1}^{n_l}$.*

Equivalently, under these assumptions, the SEM in (1) can be rewritten as follows,

$$(I - \gamma W)y_l = (I - \gamma W)X_l\beta + \varepsilon_l, \; l = 1, 2, \ldots\ldots, L. \tag{2}$$

Let $y(\gamma)_l = (I - \gamma W)y_l, X(\gamma)_l = (I - \gamma W)X_l$, Equation (2) is equivalent to

$$y(\gamma)_l = X(\gamma)_l\beta + \varepsilon_l, \quad l = 1, 2, \ldots\ldots, L, \tag{3}$$

and it can be rewritten in matrix form, as follows,

$$y(\gamma) = X(\gamma)\beta + \varepsilon. \tag{4}$$

Let $\Theta = (\beta, \sigma^2, \gamma)$. The maximum likelihood estimator (MLE) of Θ can be obtained by maximizing the concentrated likelihood function [25]. Specifically, we first fix γ and find the MLEs of β *and* σ^2 as a function of γ as follows,

$$\hat{\beta}(\gamma) = \left(X(\gamma)^T X(\gamma)\right)^{-1} X(\gamma)^T y(\gamma) \tag{5}$$

$$\hat{\sigma}^2(\gamma) = \frac{1}{n}\left(y(\gamma) - X(\gamma)\hat{\beta}(\gamma)\right)^T\left(y(\gamma) - X(\gamma)\hat{\beta}(\gamma)\right). \quad (6)$$

Secondly, we plug $\hat{\beta}$ and $\hat{\sigma}^2$ into the log-likelihood to obtain the MLE of γ. Finally, the MLEs for $\hat{\beta}$ and $\hat{\sigma}^2$ are acquired by substituting $\hat{\gamma}$ for γ in Equations (5) and (6), respectively.

Thus, when the data generation process (DGP) is a special case of (1), the following formular holds:

$$\hat{\beta}(\gamma) - \beta_0 = \left(X(\gamma)^T X(\gamma)\right)^{-1}\sum_{l=1}^{L} X(\gamma)_l^T \varepsilon_l = \left(\sum_{l=1}^{L} X(\gamma)_l^T X(\gamma)_l\right)^{-1}\sum_{l=1}^{L} s(\gamma)_l \quad (7)$$

where β_0 is the true value of β, and $s(\gamma)_l = X(\gamma)_l^T \varepsilon_l$ represents the $r \times 1$ score vector corresponding to the l^{th} cluster. Assuming that

$$E\left(s(\gamma)_l s(\gamma)_l{}^T\right) = \Sigma_l E\left(s(\gamma)_l s(\gamma)_{l'}{}^T\right) = 0 \;\; l, l' = 1, 2, \ldots\ldots, L, l \neq l', \quad (8)$$

where the expectations here are conditional on the $X(\gamma)_l$, the matrix Σ_l denotes a $r \times r$ positive semi-definite and symmetric matrix, which is the (conditional) covariance matrix of the score vector corresponding to the l^{th} cluster. Based on Equations (7) and (8), we provide a more informative representation of the (conditional) covariance matrix of $\hat{\beta}(\gamma)$, which makes it more explicit that the key to estimate is Σ_l in terms of the covariance matrix of the score vectors, and it follows

$$Var(\hat{\beta}(\gamma)) = \left(X(\gamma)^T X(\gamma)\right)^{-1}\left(\sum_{l=1}^{L} \Sigma_l\right)\left(X(\gamma)^T X(\gamma)\right)^{-1}. \quad (9)$$

When estimating Σ_l, it is natural to apply the outer products of the empirical score vectors:

$$\hat{s}(\gamma)_l = X(\gamma)_l^T \hat{\varepsilon}_l = X(\gamma)_l^T\left(y(\gamma)_l - X(\gamma)_l\hat{\beta}(\gamma)\right) = X(\gamma)_l^T y(\gamma)_l - X(\gamma)_l^T X(\gamma)_l \hat{\beta}(\gamma) \quad (10)$$

By correcting for degrees of freedom, we acquire the most widely used CRVE [24]:

$$\widehat{Var}(\hat{\beta}(\gamma)) = \frac{L(n-1)}{(L-1)(n-r)}\left(X(\gamma)^T X(\gamma)\right)^{-1}\left(\sum_{l=1}^{L} \hat{s}(\gamma)_l \hat{s}(\gamma)_l{}^T\right)\left(X(\gamma)^T X(\gamma)\right)^{-1}. \quad (11)$$

Note that given the matrix $X(\gamma)^T X(\gamma)$ and the vectors $X(\gamma)_l^T y(\gamma)_l$, $\widehat{Var}(\hat{\beta}(\gamma))$ can be acquired without computing the residual subvector $\hat{\varepsilon}_l$, which can reduce the computational costs. Since each of the $\hat{s}(\gamma)_l \hat{s}(\gamma)_l{}^T$ matrices has rank at most 1, then (11) has rank at most L (it has rank $L-1$ in many situations), which indicates that asymptotic inference derived from the above (11) may not be reliable when L is not large, and particularly when there are several restrictions. Consequently, the bootstrap methods are essential.

3. Bootstrap Computations for Maximum Likelihood Estimation of SEM

Bootstrapping is an alternative method for inferring statistical parameters where computational complexity arises [10]. In addition, different cluster bootstrap methods (to illustrate the asymptotic properties of the estimator proposed in our paper, we cite an existing theorem in Appendix A and briefly explain why this theorem is applicable to our estimator.) for SEM yield bootstrap samples in different ways, which incur different computational costs. In this part, we discuss the fast *pairs cluster bootstrap* method for SEM and the fast unrestricted and restricted wild cluster bootstrap methods for SEM, respectively.

3.1. The Fast Pairs Cluster Bootstrap Method for SEM

Bootstrapping is an alternative method for inferring statistical parameters where computational complexity arises [10]. In addition, different cluster bootstrap methods

for SEM yield bootstrap samples in different ways, which incur different computational costs. In this part, we discuss the fast *pairs cluster bootstrap* method for SEM and the fast unrestricted and restricted wild cluster bootstrap methods for SEM, respectively.

The original *pairs cluster bootstrap* method for SEM is to group the data for each cluster into a $[X(\gamma)_l,\ y(\gamma)_l]$ pair, and then to resample the data from the original L pairs with replacement, such that each bootstrap sample is composed of the original L pairs with equal probability $1/L$. The big difference between the fast *pairs cluster bootstrap* method (pcb) and the original method is the first step. As for pcb, firstly, we group the data for each cluster into a $\left[X(\gamma)_l^T X(\gamma)_l,\ X(\gamma)_l^T y(\gamma)_l\right]$ pair, and then we resample from the L pairs. Consequently, each bootstrap sample is constructed by randomly choosing L pairs with equal probability $1/L$, and the j^{th} $(j = 1, 2, \ldots\ldots B)$ bootstrap sample is represented as follows,

$$\left[X(\gamma)_l^{*j^T} X(\gamma)_l^{*j},\ X(\gamma)_l^{*j^T} y(\gamma)_l^{*j}\right],\ l = 1, 2, \ldots\ldots L. \tag{12}$$

Thus, the fast *pairs cluster bootstrap* estimates of β and $s(\gamma)_l$ are given by the following, respectively:

$$\hat{\beta}(\gamma)^{*j} = \left(\sum_{l=1}^{L} X(\gamma)_l^{*j^T} X(\gamma)_l^{*j}\right)^{-1} \sum_{l=1}^{L} X(\gamma)_l^{*j^T} y(\gamma)_l^{*j} \tag{13}$$

$$\hat{s}(\gamma)_l^{*j} = X(\gamma)_l^{*j^T} y(\gamma)_l^{*j} - X(\gamma)_l^{*j^T} X(\gamma)_l^{*j} \hat{\beta}(\gamma)^{*j}. \tag{14}$$

Typically, r and L are small, and the computational cost in both (13) and (14) is $O\left(r^2 L\right)$. Furthermore, the CRVE for $\hat{\beta}(\gamma)^{*j}$ is

$$\widehat{Var}\left(\hat{\beta}(\gamma)^{*j}\right)_{pcb} = \frac{L(n-1)}{(L-1)(n-r)} \left(\sum_{l=1}^{L} X(\gamma)_l^{*j^T} X(\gamma)_l^{*j}\right)^{-1} \left(\sum_{l=1}^{L} \hat{s}(\gamma)_l^{*j} \left(\hat{s}(\gamma)_l^{*j}\right)^T\right) \left(\sum_{l=1}^{L} X(\gamma)_l^{*j^T} X(\gamma)_l^{*j}\right)^{-1}, \tag{15}$$

and the bootstrap test statistic for $R^T\beta = 0$, where R is a known r-vector:

$$t_{pcb}{}^{*j} = \frac{R^T(\hat{\beta}(\gamma)^{*j} - \hat{\beta}(\gamma))}{\left(R^T \hat{Var}\left(\hat{\beta}(\gamma)^{*j}\right)_{pcb} R\right)^{1/2}}. \tag{16}$$

Let $c^*_{\frac{\alpha}{2}}$ and $c^*_{1-\frac{\alpha}{2}}$ represent the $\frac{\alpha}{2}$ and $1-\frac{\alpha}{2}$ quantiles of the $t_{pcb}{}^{*j}$, respectively, and based on these statistics, the bootstrap percentile-t confidence interval for β is constructed as follows,

$$\left[\hat{\beta}(\gamma) - s\left(\hat{\beta}(\gamma)\right) c^*_{1-\frac{\alpha}{2}},\ \hat{\beta}(\gamma) - s\left(\hat{\beta}(\gamma)\right) c^*_{\frac{\alpha}{2}}\right], \tag{17}$$

where $s\left(\hat{\beta}(\gamma)\right)$ is the cluster-robust standard error of $\hat{\beta}(\gamma)$.

In terms of the pcb, there are several limitations. Firstly, the pcb generally involves higher computational costs; specifically, the matrix $\sum_{l=1}^{L} X(\gamma)_l^{*j^T} X(\gamma)_l^{*j}$ in the above calculation is different for each bootstrap sample and requires to be generated and inverted B times. Secondly, a technical problem that is typically ignored is that it is possible for $\sum_{l=1}^{L} X(\gamma)_l^{*j^T} X(\gamma)_l^{*j}$ to be singular in a simulated bootstrap sample, in which case the least squares estimator $\hat{\beta}(\gamma)^{*j}$ cannot be defined, especially when our problem involves treatment at the level of the cluster and few clusters are treated, and $\hat{\beta}(\gamma)^{*j}$ obtained based on this method may be inaccurate [5,6,26,27]. Finally, since the null hypothesis is not imposed on bootstrap samples, many of the existing studies indicate that pcb may not yield satisfactory results, especially for samples with different sizes [19].

3.2. The Fast Wild Cluster Bootstrap Methods for SEM

The original *wild cluster bootstrap* methods use auxiliary random variables v_l^* for each cluster, which are i.i.d. with zero mean and unit variance, and the most popular choice for v_l^* is the Rademacher random variables, which satisfy $P(v_l^* = 1) = \frac{1}{2}$ and $P(v_l^* = -1) = \frac{1}{2}$ [8,16,18,28]. The bootstrap observations are then generated as $y(\gamma)_l{}^* = X(\gamma)_l\hat{\beta}(\gamma) + \varepsilon_l{}^*$ with $\varepsilon_l{}^* = \hat{\varepsilon}_l v_l^*$ $(l = 1, 2, \ldots\ldots, L)$, where the regressors $X(\gamma)_l$ are held fixed at their sample values, $\hat{\beta}(\gamma)$ denotes the sample estimator, and $\hat{\varepsilon}_l$ are the least squares residuals, which are also held fixed at their sample values. The big difference between fast *wild cluster bootstrap* methods (WCB) and the original method is the first step. As for WCB, the bootstrap disturbances affect the estimates only through scores with $s(\gamma)_l^* = \hat{s}(\gamma)_l v_l^*$, and based on whether generating bootstrap samples depends on the restrictions to be tested, WCB is divided into the fast unrestricted and restricted wild cluster bootstrap methods.

3.2.1. The Fast Unrestricted Wild Cluster Bootstrap Method for SEM

The fast *unrestricted wild cluster bootstrap* method (UWC) generates bootstrap samples that do not depend on the restrictions to be tested; furthermore, it is based on different steps of the computation, which is divided into UWC1, UWC2, and UWC3.

The detailed steps for applying UWC1 to generate the j^{th} bootstrap sample and statistics are described below:

Step 1: Generate v_l^{*j} from Rademacher distribution, and furthermore, generate the bootstrap score vectors as follows,

$$s(\gamma)_l^{*j} = v_l^{*j}\hat{s}(\gamma)_l,\ l = 1, 2, \ldots\ldots L, \tag{18}$$

where $\hat{s}(\gamma)_l = X(\gamma)_l^T y(\gamma)_l - X(\gamma)_l^T X(\gamma)_l\hat{\beta}(\gamma)$.

Step 2: Based on the bootstrap scores from (18), we can easily obtain

$$\hat{\beta}(\gamma)^{*j} - \hat{\beta}(\gamma) = \left(X(\gamma)^T X(\gamma)\right)^{-1}\sum_{l=1}^{L} s(\gamma)_l^{*j}. \tag{19}$$

And Equation (10) is equivalent to

$$\hat{s}(\gamma)_l = s(\gamma)_l - X(\gamma)_l^T X(\gamma)_l(\hat{\beta}(\gamma) - \beta_0)\ , \qquad l = 1, 2, \ldots\ldots L \tag{20}$$

Similarly, the j^{th} empirical bootstrap score $\hat{s}(\gamma)_l^{*j}$ are given as follows,

$$\hat{s}(\gamma)_l^{*j} = s(\gamma)_l^{*j} - X(\gamma)_l^T X(\gamma)_l\left(\hat{\beta}(\gamma)^{*j} - \hat{\beta}(\gamma)\right), \qquad l = 1, 2, \ldots\ldots L. \tag{21}$$

Step 3: We can obtain the CRVE for $\hat{\beta}(\gamma)^{*j}$ and the bootstrap test statistic for $R^T\beta = 0$, respectively:

$$\hat{Var}\left(\hat{\beta}(\gamma)^{*j}\right)_{UWC1} = \frac{L(n-1)}{(L-1)(n-r)}\left(X(\gamma)^T X(\gamma)\right)^{-1}\left(\sum_{l=1}^{L}\hat{s}(\gamma)_l^{*j}\left(\hat{s}(\gamma)_l^{*j}\right)^T\right)\left(X(\gamma)^T X(\gamma)\right)^{-1} \tag{22}$$

$$t_{UWC1}{}^{*j} = \frac{R^T(\hat{\beta}(\gamma)^{*j} - \hat{\beta}(\gamma))}{\left(R^T\hat{Var}\left(\hat{\beta}(\gamma)^{*j}\right)_{UWC1}R\right)^{1/2}}\ . \tag{23}$$

Similar to the bootstrap percentile-t confidence interval constructed by pcb in Section 3.1, we can acquire the bootstrap percentile-t confidence interval constructed by UWC1.

UWC1 involves some unnecessary work in order to obtain t-statistics and further construct confidence intervals, which can be improved by UWC2.

In fact, when we calculate the numerator of the t-statistic, it is possible to avoid calculating $\hat{\beta}(\gamma)^{*j}$. To this end, before the bootstrap loop, form the matrices

$$G(\gamma)_l = X(\gamma)_l^T X(\gamma)_l \left(X(\gamma)^T X(\gamma) \right)^{-1}, \qquad l = 1, 2, \ldots\ldots L. \tag{24}$$

generate the vector

$$v_j^* = \left(v_1^{*j}, \ldots\ldots, v_L^{*j} \right), \tag{25}$$

and compute the numerator of the t-statistic as follows,

$$\begin{aligned} R^T \left(\hat{\beta}(\gamma)^{*j} - \hat{\beta}(\gamma) \right) &= \sum_{l=1}^{L} R^T \left(X(\gamma)^T X(\gamma) \right)^{-1} s(\gamma)_l^{*j} = \sum_{l=1}^{L} R^T \left(X(\gamma)^T X(\gamma) \right)^{-1} \hat{s}(\gamma)_l v_l^{*j} \\ &= \sum_{l=1}^{L} f(\gamma)_l v_l^{*j} = f(\gamma)^T v_j^* , \end{aligned} \tag{26}$$

where

$$f(\gamma)_l \equiv R^T \left(X(\gamma)^T X(\gamma) \right)^{-1} \hat{s}(\gamma)_l$$

and

$$f(\gamma) \equiv (f(\gamma)_1, \ldots\ldots, f(\gamma)_L). \tag{27}$$

The detailed steps for applying UWC2 to generate the j^{th} bootstrap sample and statistics are described below.

Step 1: Generate the bootstrap score vectors as follows,

$$s(\gamma)_l^{*j} = v_l^{*j} \hat{s}(\gamma)_l, \; l = 1, 2, \ldots\ldots L, \tag{28}$$

where $\hat{s}(\gamma)_l = X(\gamma)_l^T y(\gamma)_l - X(\gamma)_l^T X(\gamma)_l \hat{\beta}(\gamma)$.

Step 2: Based on Equation (24), we can obtain the j^{th} empirical bootstrap scores $\hat{s}(\gamma)_l^{*j}$ as follows,

$$\hat{s}(\gamma)_l^{*j} = s(\gamma)_l^{*j} - G(\gamma)_l \textstyle\sum_{g=1}^{L} s(\gamma)_g^{*j}, \qquad l = 1, 2, \ldots\ldots L. \tag{29}$$

Step 3: We can calculate the CRVE for $\hat{\beta}(\gamma)^{*j}$ and the bootstrap t-statistic, respectively:

$$\begin{aligned} &\widehat{Var} \left(\hat{\beta}(\gamma)^{*j} \right)_{UWC2} \\ &= \tfrac{L(n-1)}{(L-1)(n-r)} \left(X(\gamma)^T X(\gamma) \right)^{-1} \left(\sum_{l=1}^{L} \hat{s}(\gamma)_l^{*j} \left(\hat{s}(\gamma)_l^{*j} \right)^T \right) \left(X(\gamma)^T X(\gamma) \right)^{-1} \end{aligned} \tag{30}$$

$$t_{UWC2}{}^{*j} = \frac{R^T (\hat{\beta}(\gamma)^{*j} - \hat{\beta}(\gamma))}{\left(R^T \widehat{Var} \left(\hat{\beta}(\gamma)^{*j} \right)_{UWC2} R \right)^{1/2}} \quad . \tag{31}$$

In fact, in terms of calculating the denominator of the bootstrap t-statistic, a trick proposed by Roodman et al. [29] can further reduce the computational cost, and based on this trick, a method called UWC3 is derived. Specifically, before the bootstrap loop, in addition to (24), (25), (26), and (27), we can define the matrix $C(\gamma)$ with the typical element:

$$C(\gamma)_{lg} = R^T \left(X(\gamma)^T X(\gamma) \right)^{-1} X(\gamma)_l^T X(\gamma)_l \left(X(\gamma)^T X(\gamma) \right)^{-1} \hat{s}(\gamma)_g. \tag{32}$$

The detailed steps for applying UWC3 to generate the j^{th} bootstrap sample and statistics are described below.

Step 1: For each bootstrap replication, based on (25) and (27), we can calculate the q_l^{*j} instead of $\hat{s}(\gamma)_l^{*j}$ in the denominator of the bootstrap t-statistic for UWC2 as follows,

$$q_l^{*j} = f(\gamma)_l v_l^{*j} - \sum_{g=1}^{L} C(\gamma)_{lg} v_g^{*j}, \quad l = 1, 2, \ldots\ldots L. \tag{33}$$

Step 2: We can obtain the denominator of the bootstrap t-statistic and the bootstrap t-statistic, respectively, as follows,

$$R^T \hat{Var}\left(\hat{\beta}(\gamma)^{*j}\right) R = \frac{L(n-1)}{(L-1)(n-r)} \sum_{l=1}^{L} \left(q_l^{*j}\right)^2 \tag{34}$$

$$t_{UWC3}{}^{*j} = \frac{R^T(\hat{\beta}(\gamma)^{*j} - \hat{\beta}(\gamma))}{\left(R^T \hat{Var}\left(\hat{\beta}(\gamma)^{*j}\right) R\right)^{1/2}} \ . \tag{35}$$

Furthermore, we can obtain the bootstrap percentile-t confidence interval constructed by UWC2 and UWC3, respectively, which are similar to that of UWC1.

Obviously, by applying Equation (18), UWCs have two advantages over pcb; firstly, the computational cost of applying Equation (18) is $O\ (rL)$, rather than $O((r+\ 1)\ n)$ that would be required by the traditional methods [30]. Secondly, Equation (18) can preserve the covariance matrix of the scores evidently; specifically, $\hat{\beta}(\gamma)^{*j}$ is determined by the bootstrap scores, while the covariance matrix is determined by the empirical bootstrap scores, which captures the key feature of WCB.

Moreover, when calculating the denominator of the bootstrap t-statistic, the computational cost required to form $\hat{s}(\gamma)_l^{*j}$ in UWC1 or UWC2 is $O\ \left(r^2 L\right)$. Meanwhile, when the initial work of UWC3 is completed, the effort required for each bootstrap sample is $O\ \left(L^2\right)$ and does not rely on n or r. Therefore, in general, UWC3 is less computationally expensive than UWC2, except for very large L.

3.2.2. The Fast Restricted Wild Cluster Bootstrap Method for SEM

The difference between the *restricted wild cluster bootstrap* method (RWC) and UWC is that the former uses the restricted empirical score vectors $\widetilde{s}(\gamma)_l$ instead of unrestricted ones $\hat{s}(\gamma)_l$.

Let us partition β as $[\beta_1,\ \beta_2]$, where β_1 is a $(r\ -1)$-vector, and β_2 is a scalar, and consider a restriction of the form $\beta_2 = 0$. When $X(\gamma)_l$ is partitioned conformably with β, the SEM can be rewritten as

$$y(\gamma)_l = X(\gamma)_{1l}\beta_1 + \beta_2 x(\gamma)_{2l} + \varepsilon_l\ , \qquad l = 1, 2, \ldots\ldots L, \tag{36}$$

where $X(\gamma)_{1l}$ represents the $n_l \times (r-1)$ matrix, and $x(\gamma)_{2l}$ denotes the n_l-vector, with $X(\gamma)_l = [X(\gamma)_{1l}, x(\gamma)_{2l}]$. It can also be rewritten in matrix form as follows,

$$y(\gamma) = X(\gamma)_1\beta_1 + \beta_2 x(\gamma)_2 + \varepsilon\ ,$$

and the restrict MLEs of β is given by

$$\widetilde{\beta}(\gamma) = \begin{bmatrix} \left(X(\gamma)_1{}^T X(\gamma)_1\right)^{-1} X(\gamma)_1{}^T y(\gamma) \\ 0 \end{bmatrix},$$

where $X(\gamma)_1{}^T X(\gamma)_1$ consists of the $(r\ -1)\ \times\ (r-1)$ upper-left matrix block of $X(\gamma)^T X(\gamma)$, and $X(\gamma)_1{}^T y(\gamma)$ consists of the first $(r-1)$ elements of $X(\gamma)^T y(\gamma)$.

Before the bootstrap loop, form the matrix $\widetilde{C}(\gamma)$ with the typical element,

$$\widetilde{C}(\gamma)_{lg} = R^T\left(X(\gamma)^T X(\gamma)\right)^{-1} X(\gamma)_l^T X(\gamma)_l \left(X(\gamma)^T X(\gamma)\right)^{-1} \widetilde{s}(\gamma)_g\,, \tag{37}$$

generate the vector

$$\widetilde{v}_j^* \equiv \left(\widetilde{v}_1^{*j}, \ldots\ldots, \widetilde{v}_L^{*j}\right), \tag{38}$$

and compute the numerator of the t-statistic as follows,

$$R^T\left(\widetilde{\beta}(\gamma)^{*j} - \widetilde{\beta}(\gamma)\right) = \sum_{l=1}^{L} R^T\left(X(\gamma)^T X(\gamma)\right)^{-1} \widetilde{s}(\gamma)_l \widetilde{v}_l^{*j} = \sum_{l=1}^{L} \widetilde{f}(\gamma)_l \widetilde{v}_l^{*j} = \widetilde{f}(\gamma)^T \widetilde{v}_j^*\,, \tag{39}$$

where

$$\begin{aligned} \widetilde{f}(\gamma)_l &\equiv R^T\left(X(\gamma)^T X(\gamma)\right)^{-1} \widetilde{s}(\gamma)_l, \\ \widetilde{f}(\gamma) &\equiv \left(\widetilde{f}(\gamma)_1, \ldots\ldots, \widetilde{f}(\gamma)_L\right), \end{aligned} \tag{40}$$

and $\widetilde{s}(\gamma)_l = X(\gamma)_l^T y(\gamma)_l - X(\gamma)_l^T X(\gamma)_l \widetilde{\beta}(\gamma)$.

The detailed steps for applying RWC to generate the j^{th} bootstrap sample and statistics are described below.

Step 1: For each bootstrap replication, we can calculate the $\widetilde{q}_l^{*j}$ in the denominator of the bootstrap t-statistic as follows,

$$\widetilde{q}_l^{*j} = \widetilde{f}(\gamma)_l \widetilde{v}_l^{*j} - \textstyle\sum_{g=1}^{L} \widetilde{C}(\gamma)_{lg} \widetilde{v}_g^{*j}, l = 1, 2, \ldots\ldots L. \tag{41}$$

Step 2: We can obtain the denominator of the bootstrap t-statistic and the bootstrap t-statistic, respectively:

$$R^T \widehat{Var}\left(\widetilde{\beta}(\gamma)^{*j}\right) R = \frac{L(n-1)}{(L-1)(n-r)} \textstyle\sum_{l=1}^{L} \left(\widetilde{q}_l^{*j}\right)^2 \tag{42}$$

$$t_{RWC}{}^{*j} = \frac{R^T\left(\widetilde{\beta}(\gamma)^{*j} - \widetilde{\beta}(\gamma)\right)}{\left(R^T \widehat{Var}\left(\widetilde{\beta}(\gamma)^{*j}\right) R\right)^{1/2}}\,. \tag{43}$$

However, the bootstrap percentile-t confidence interval constructed by RWC differs from the one by UWC. Typically, inverting a RWC bootstrap test is a good way to obtain a confidence interval for one of the parameters in a linear model. In specific, the confidence interval of the parameter is obtained by performing two RWC bootstrap tests: one for the upper limit of the interval, and the other for the lower limit of the interval [30,31]. For example, if we intend to form a bootstrap confidence interval for the parameter β_2 based on RWC, it is necessary to acquire the upper and lower limits of the interval through an iterative process. Specifically, let $\beta(\gamma)_2^0$ represent one of the limits of the confidence interval, the upper limit $\beta(\gamma)_2^u$ or the lower limit $\beta(\gamma)_2^l$, and in such a case, we could apply the equal-tail bootstrap p-value as follows,

$$\hat{P}_{et}^*\left(\beta(\gamma)_2^0\right) = 2min\left(\frac{1}{B}\sum_{j=1}^{B} \mathbb{I}\left(t_2^{*j} > t_2\right), \frac{1}{B}\sum_{j=1}^{B} \mathbb{I}\left(t_2^{*j} \le t_2\right)\right),$$

where

$$t_2 = \frac{\widetilde{\beta}(\gamma)_2 - \beta(\gamma)_2^0}{\left(\hat{Var}\left(\widetilde{\beta}(\gamma)_2\right)\right)^{1/2}},\ t_2^{*j} = \frac{\widetilde{\beta}(\gamma)_2^{*j} - \beta(\gamma)_2^0}{\left(\hat{Var}\left(\widetilde{\beta}(\gamma)_2^{*j}\right)\right)^{1/2}},\ j = 1, \ldots\ldots, B, \tag{44}$$

and $\mathbb{I}(\cdot)$ is an indicator function that takes 1 if its argument is true and 0 otherwise.

Let α denotes the significance level. Obviously, the lower limit $\beta(\gamma)_2^l$ is virtually certain to be less than $\widetilde{\beta}(\gamma)_2$; thus, $\hat{P}_{et}^*\left(\beta(\gamma)_2^l\right)$ is less than α for $\beta(\gamma)_2^l > \beta_2$ and greater than α for $\beta(\gamma)_2^l < \beta_2$. Similarly, it is certain that $\beta(\gamma)_2^u$, which is greater than $\widetilde{\beta}(\gamma)_2$, also satisfies these two inequalities.

During the iterative process, to determine the bootstrap confidence interval of β_2 based on RWC, $\hat{P}_{et}^*\left(\beta(\gamma)_2^0\right)$ is needed to be evaluated several times with the same set of realized values of the auxiliary random variables. Specifically, before generating the bootstrap samples and statistics for any value $\beta(\gamma)_2^0$, we need to form the vector $\widetilde{f}(\gamma)$ and the matrix $\widetilde{C}(\gamma)$ for testing $\beta_2 = \beta(\gamma)_2^0$, respectively, which both depend on $\beta(\gamma)_2^0$. Let $h(\gamma) = \left(X(\gamma)^T X(\gamma)\right)^{-1} R$ and $h'(\gamma) = \left(X(\gamma)^T X(\gamma)\right)^{-1} X(\gamma)_l{}^T X(\gamma)_l h(\gamma)$, where R is a vector with 1 for the r^{th} element and 0 for other elements, then $\widetilde{f}(\gamma)_l$ and $\widetilde{C}(\gamma)_{lg}$ can be rewritten as

$$\begin{aligned}\widetilde{f}(\gamma)_l\left\{\beta(\gamma)_2^0\right\} = {} & y(\gamma)_l{}^T X(\gamma)_l h(\gamma) - y(\gamma)^T X(\gamma)_1 \left(X(\gamma)_1{}^T X(\gamma)_1\right)^{-1} X(\gamma)_{1l}{}^T X(\gamma)_l h(\gamma) - \\ & \beta(\gamma)_2^0 \left(x(\gamma)_{2l}{}^T X(\gamma)_l h(\gamma) - x(\gamma)_2{}^T X(\gamma)_1 \left(X(\gamma)_1{}^T X(\gamma)_1\right)^{-1} X(\gamma)_{1l}{}^T X(\gamma)_l h(\gamma)\right)\end{aligned} \tag{45}$$

and

$$\begin{aligned}\widetilde{C}(\gamma)_{lg}\left\{\beta(\gamma)_2^0\right\} = {} & y(\gamma)_l{}^T X(\gamma)_l h'(\gamma) \\ & - y(\gamma)^T X(\gamma)_1 \left(X(\gamma)_1{}^T X(\gamma)_1\right)^{-1} X(\gamma)_{1l}{}^T X(\gamma)_l h'(\gamma) \\ & - \beta(\gamma)_2^0 \left(x(\gamma)_{2l}{}^T X(\gamma)_l h'(\gamma)\right. \\ & \left. - x(\gamma)_2{}^T X(\gamma)_1 \left(X(\gamma)_1{}^T X(\gamma)_1\right)^{-1} X(\gamma)_{1l}{}^T X(\gamma)_l h'(\gamma)\right) .\end{aligned} \tag{46}$$

Moreover, for each bootstrap sample, the bootstrap t-statistic for testing $\beta_2 = \beta(\gamma)_2^0$ is available based on (39), (41), and (42), and it involves less computational cost. However, since $\hat{P}_{et}^*\left(\beta(\gamma)_2^0\right)$ is not a continuous function of its parameter, it is not possible to obtain the optimal value using the methods that rely on derivatives. Based on the properties discussed above, the bisection method is a reliable, easy to implement, and guaranteed convergence method to determine the real roots of such problems, and the steps for applying the bisection method to determine the limits of the confidence interval are as follows,

Step1: Define $f\left(\beta(\gamma)_2^0\right) = \hat{P}_{et}^*\left(\beta(\gamma)_2^0\right) - \alpha$, choose initial values of β_2, including $\beta(\gamma)_2^a$ and $\beta(\gamma)_2^b$, such that $f\left(\beta(\gamma)_2^b\right) f\left(\beta(\gamma)_2^a\right) < 0$, and tolerance rate e.

Step 2: If $f\left(\beta(\gamma)_2^b\right) f\left(\beta(\gamma)_2^a\right) \geq 0$, then the root does not lie in this interval.

Step 3: Find the midpoint, set $\beta(\gamma)_2^m = \left(\beta(\gamma)_2^b + \beta(\gamma)_2^a\right)/2$,

(i) if the function value of the midpoint $f\left(\beta(\gamma)_2^m\right) = 0$, then $\beta(\gamma)_2^m$ is the root. Go to Step 5.

(ii) if $f\left(\beta(\gamma)_2^b\right) f(\beta(\gamma)_2^m) < 0$, the root lies between $\beta(\gamma)_2^b$ and $\beta(\gamma)_2^m$, then set $\beta(\gamma)_2^b = \beta(\gamma)_2^b$, $\beta(\gamma)_2^a = \beta(\gamma)_2^m$.

(iii) or else set $\beta(\gamma)_2^b = \beta(\gamma)_2^m$, $\beta(\gamma)_2^a = \beta(\gamma)_2^a$.

Step 4: If the $\left|\beta(\gamma)_2^b - \beta(\gamma)_2^a\right|$ is higher than e, go to step 3, otherwise display the interval $\left[\beta(\gamma)_2^a, \beta(\gamma)_2^b\right]$, where $\hat{P}_{et}^*\left(\beta(\gamma)_2^a\right) \cong \alpha$ and $\hat{P}_{et}^*\left(\beta(\gamma)_2^b\right) \cong \alpha$.

Step 5: Display $\beta(\gamma)_2^m$ as the approximate root, which satisfies $\hat{P}_{et}^*\left(\beta(\gamma)_2^m\right) = \alpha$.

The bisection process is repeated, and finally, the interval $\left[\beta(\gamma)_2^l, \beta(\gamma)_2^u\right]$ is available.

4. Computing Costs

Obviously, the methods presented in Section 3 are much faster than the bootstrap method that directly generates the full bootstrap samples. Specifically, if we perform the full maximum likelihood estimation of SEM for each bootstrap sample, the cost of each bootstrap replication is $O\left(r^2\, n\right)$. In contrast, by pre-computing some sufficient statistics, we can calculate all of the quantities needed for bootstrap inference without performing a computationally expensive process of $O\left(n\right)$. Thus, all calculations with computational cost $O\left(n\right)$ are performed only once, not for each bootstrap sample, and the computational cost of each bootstrap replication depends solely on L and r, but not on n.

In this section, we support the above claims by simulating the computational cost of calculating the bootstrap t-statistics with pcb, UWC1, UWC2, UWC3, and RWC, as proposed in Section 3, respectively. In addition, for comparative purposes, this study also reports a benchmark number that is $B + 1$ times the computing time of a single test statistic. Table 2 illustrates the average computing times (in seconds) of 1000 Monte Carlo simulations for various bootstrap methods with $n = 1000$ or $100,000$, $L = 10$ or 20, $B = 999$ or 9999, $r = 10$, γ varying in the set $\{0.2,\ 0.4,\ 0.6,\ 0.8\}$, and W having the rook-based contiguity neighborhood. In addition, the consuming time to generate the data (which is typically larger than the computing time of bootstrapping) is not included in all experimental results.

Table 2. Computing times (in seconds) for various bootstrap methods.

$n = 1000$				
$L,\ B$	$10, 999$	$20, 999$	$10, 9999$	$20, 9999$
Benchmark	0.1844	0.2003	1.8440	2.0030
pcb	0.0895	0.0899	0.8850	0.8889
UWC1	0.0057	0.0068	0.0560	0.0645
UWC2	0.0046	0.0052	0.0450	0.0517
UWC3	0.0017	0.0023	0.0169	0.0184
RWC	0.0012	0.0016	0.0103	0.0110
$n = 100,000$				
$L,\ B$	$10, 999$	$20, 999$	$10, 9999$	$20, 9999$
Benchmark	20.3400	21.5200	203.4000	215.2000
pcb	0.8610	0.8740	1.0103	1.0216
UWC1	0.0614	0.0620	0.1052	0.1083
UWC2	0.0429	0.0435	0.0856	0.0867
UWC3	0.0174	0.0187	0.0291	0.0293
RWC	0.0102	0.0108	0.0194	0.0195

Firstly, it is obvious that pcb is much less computationally expensive than the baseline method; nevertheless, pcb is more computationally expensive than UWCs since UWCs pre-compute the same $X(\gamma)^T X(\gamma)$ matrix for all bootstrap samples, while pcb involves constructing the ${X(\gamma)^{*j}}^T X(\gamma)^{*j}$ matrix for each bootstrap sample. Secondly, compared to UWC1, UWC2 is faster, which may be attributed to the fact that the $G(\gamma)_l$ matrices and

$f(\gamma)$ and v_j^* vectors are formed before the bootstrap loop, which makes computing the t-statistic less expensive. Moreover, UWC3 applies a trick to compute the denominator of the bootstrap t-statistic, which further reduces computational cost. Thirdly, applying RWC is marginally faster than UWC3, and it performs the fastest; therefore, it can be used for large samples in applications. Finally, when $n = 100,000$, increasing the times of B has less impact on the computational cost, which is different from the performance when $n = 1000$, and this is because that the initial computation takes up a large proportion of the total time for large samples.

5. Monte Carlo Simulations

To demonstrate our theoretical findings, we perform 1000 Monte Carlo simulations and 999 bootstrap iterations to investigate the empirical coverage frequencies of nominal 95% confidence intervals constructed by pcb, UWC1, UWC2, UWC3, and RWC, respectively. Specifically, the Monte Carlo analysis was performed with a simple SEM, which includes one explanatory variable and a constant:

$$(I - \gamma W)y = (I - \gamma W)X\beta_1 + \beta_0 + \varepsilon,$$

where $\beta_1 = 0.2$, $\beta_0 = 0.5$, ε is generated from a normal distribution with zero mean and unit variance, and X is drawn from another normal distribution with zero mean and variance 2. We consider a rook-based contiguity neighborhood for the spatial weight matrix W, which has been normalized, and the parameter γ varies over the set $\{0.2, 0.4, 0.6, 0.8\}$.

In each of these experiments, there are a total of $n = 100L$ observations, and these observations are grouped into L clusters given by the following equations,

$$\begin{gathered} n_l = \left[n \frac{e^{\frac{\delta l}{L}}}{\sum_{h=1}^{L} e^{\frac{\delta h}{L}}} \right], l = 1, 2, \ldots\ldots, L-1, \\ n_L = n - \sum_{l=1}^{L-1} n_l, \end{gathered} \tag{47}$$

where $[\cdot]$ denotes the integer of its argument. The parameter δ determines the heterogeneity of the cluster size; specifically, when $\delta = 0$, for all l there holds $n_l = n/L$, while the difference in cluster size grows as δ increases. Table 3 reports the two-sided equal-tailed coverage frequencies of β_1 with the precisions of the bootstrap confidence intervals (in parentheses) when $L = 10$ and $\delta = 0$ or 3.

Table 3. Two-sided equal-tailed coverage frequencies of β_1 with 10 clusters.

	pcb		**UWC1**		**UWC2**		**UWC3**		**RWC**	
δ	**0**	**3**	**0**	**3**	**0**	**3**	**0**	**3**	**0**	**3**
$\gamma = 0.2$	0.8159	0.7855	0.8835	0.8119	0.8874	0.8224	0.8907	0.8523	0.8978	0.8728
	(0.9016)	(0.9184)	(0.4686)	(0.6031)	(0.4652)	(0.5601)	(0.4285)	(0.4916)	(0.2017)	(0.2210)
$\gamma = 0.4$	0.8231	0.7922	0.8948	0.8605	0.8980	0.8561	0.9032	0.8702	0.9302	0.8939
	(0.8550)	(0.8657)	(0.3322)	(0.4037)	(0.3370)	(0.3928)	(0.2355)	(0.3831)	(0.1911)	(0.2146)
$\gamma = 0.6$	0.8796	0.8509	0.9064	0.8829	0.9183	0.8834	0.9299	0.9194	**0.9345**	0.9255
	(0.7283)	(0.7502)	(0.2817)	(0.3932)	(0.2756)	(0.3898)	(0.2284)	(0.3205)	**(0.1814)**	(0.2036)
$\gamma = 0.8$	0.8398	0.8059	0.8829	0.8635	0.8972	0.8727	0.9030	0.8979	0.9296	0.9107
	(0.8345)	(0.8414)	(0.3247)	(0.4750)	(0.3173)	(0.4622)	(0.3031)	(0.4313)	(0.2269)	(0.2261)

From Table 3, we can find that firstly, when $\delta = 0$, (i.e., all clusters contain 100 observations), the two-sided equal-tailed coverage frequencies and the precisions of the bootstrap confidence intervals of β_1 obtained with each method perform better than $\delta = 3$ (i.e., the cluster sizes vary from 18 to 277). Secondly, the bootstrap confidence intervals of

β_1 constructed by pcb severely under-cover for all values of γ, and the UWCs improve on the pcb but still under-cover for all γ. Thirdly, the RWC outperforms all UWCs but still gives coverage less than the nominal frequency for all values of γ, which is consistent with the findings that the under-coverage of bootstrap confidence interval based on UWCs is more severe than that of bootstrap confidence interval based on RWC in some cases [21,29]. Finally, it is obvious that the RWC obtains the optimal performance with $\delta = 0$ and γ = 0.6; in particular, the two-sided equal-tailed coverage frequency of β_1 is 0.9345.

Table 4 shows the two-sided equal-tailed coverage frequencies of β_1 with the precisions of the bootstrap confidence intervals (in parentheses) when $L = 20$ and $\delta = 0$ or 3.

Table 4. Two-sided equal-tailed coverage frequencies of β_1 with 20 clusters.

	pcb		**UWC1**		**UWC2**		**UWC3**		**RWC**	
δ	**0**	**3**	**0**	**3**	**0**	**3**	**0**	**3**	**0**	**3**
$\gamma = 0.2$	0.8782 (0.7248)	0.8660 (0.7690)	0.9042 (0.2249)	0.8959 (0.2417)	0.9231 (0.2149)	0.9162 (0.2244)	0.9356 (0.2279)	0.9324 (0.2286)	0.9404 (0.1896)	0.9387 (0.1920)
$\gamma = 0.4$	0.9125 (0.5845)	0.8801 (0.6149)	0.9226 (0.2057)	0.9180 (0.2243)	0.9492 (0.1942)	0.9275 (0.2188)	0.9494 (0.1986)	0.9398 (0.2049)	**0.9612** **(0.1414)**	0.9574 (0.1482)
$\gamma = 0.6$	0.9073 (0.4742)	0.8944 (0.5015)	0.9479 (0.1860)	0.9351 (0.2004)	0.9524 (0.1805)	0.9460 (0.1904)	0.9575 (0.1811)	0.9451 (0.1968)	0.9595 (0.1515)	0.9498 (0.1546)
$\gamma = 0.8$	0.8961 (0.4894)	0.8826 (0.5120)	0.9207 (0.2037)	0.9196 (0.2139)	0.9398 (0.1987)	0.9342 (0.2105)	0.9446 (0.2003)	0.9316 (0.2106)	0.9523 (0.1602)	0.9406 (0.1639)

Firstly, with a bigger cluster size, the two-sided equal-tailed coverage frequencies and the precisions of the bootstrap confidence intervals of β_1 based on each method with $\delta = 0$ perform better than those with $\delta = 3$ (i.e., the cluster sizes vary from 16 to 304). Secondly, when $L = 20$, the results of all experiments largely outperform those when $L = 10$; in particular, the optimal coverage of β_1 based on RWC is around 0.96 with $\delta = 0$ and $\gamma = 0.4$, which is higher than the nominal frequency of 0.95. Finally, the best performances of β_1 based on UWCs and RWC are obtained with $\gamma = 0.6$ and $\gamma = 0.4$, respectively, which suggests that choosing an optimal value of γ for different methods may result in coverage closer to the nominal frequency.

6. Conclusions

This study proposes fast cluster bootstrap methods for SEM, which significantly reduce the computational cost and improve the reliability of parameter inference for SEM with large samples compared with traditional bootstrap methods. Specifically, before performing the bootstrap loop of SEM, the sufficient statistics are computed for each of the clusters, and based on these sufficient statistics, we can calculate all of the quantities needed for bootstrap inference. In addition, the test statistics and the bootstrap confidence intervals are associated with the samples solely by these sufficient statistics.

This study simulates the computational cost of calculating bootstrap t-statistics with various bootstrap methods, and the result reveals that the computational cost of our proposed methods is substantially reduced compared with traditional bootstrap methods, particularly with a large sample size.

Moreover, an extensive Monte Carlo simulation study indicates that, firstly, when all clusters have the same size, the two-sided equal-tailed coverage frequencies and the precisions of the bootstrap confidence intervals of the parameter obtained with the various methods outperform those of the clusters with varied sizes. Secondly, when $L = 20$, the results of all experiments largely outperform those when $L = 10$, and the optimal coverage of the parameter based on RWC is higher than the nominal frequency of 0.95. Finally,

choosing an optimal value of γ for different methods may result in coverage closer to the nominal frequency. Table 5 shows the main empirical findings of this study.

Table 5. The main empirical findings of this study.

The Main Empirical Findings of This Study	
Finding 1:	The computational cost of our proposed methods is substantially reduced compared with traditional bootstrap methods.
Finding 2:	The optimal coverage of the parameter based on RWC is higher than the nominal frequency.

7. Future Work

This study proposes fast cluster bootstrap methods for SEM and has the future research work as follows: firstly, our methods can be extended to other spatial models such as the Spatial Lag Model (SLM). In specific, the SLM can be written as $y_l = \theta W y_l + X_l\beta + \varepsilon_l, (l = 1, 2, \ldots\ldots, L)$. Equivalently, the SLM can be expressed as $(I - \theta W)y_l = X_l\beta + \varepsilon_l, (l = 1, 2, \ldots\ldots, L)$. Furthermore, let $y(\theta)_l = (I - \theta W)y_l$, and the SLM is equivalent to $y(\theta)_l = X_l\beta + \varepsilon_l,\ (l = 1, 2, \ldots\ldots, L)$. Similarly, it can be rewritten in matrix form with $y(\theta) = X\beta + \varepsilon$. Obviously, the bootstrap computations for maximum likelihood estimation of SLM, including the fast *pairs cluster bootstrap* method for SLM and the fast unrestricted and restricted wild cluster bootstrap methods for SLM, are similar to the calculation process used for SEM in the above sections.

Secondly, our experiments are limited in scope. We suppose that the error of the model with the classical assumption has a normal distribution. In fact, more realistic data-generating processes (e.g., non-normal errors, spatial heteroscedasticity) have more challenging problems, which will be our focus of research in the future.

Author Contributions: Y.Z. helped with conceptualization, methodology, software, investigation, and writing—original draft. H.F. was involved in submitting, writing—review and editing, and validation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (grant no. 72472150), and Fundamental Research Funds for the Central Universities (grant no. 292024082).

Data Availability Statement: Data is contained within the article. The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

We introduce the following Theorem 1 [32] to illustrate the theoretical properties of the methods proposed in this study.

Theorem A1. *Let* $X_1, X_2, \ldots\ldots$ *.be a stationary sequence of real random variables with mar ginal distribution* P*, and let* $\mathcal{F}$ *be a class of functions in* $L_2(P)$*. Also, assume that* $X_1^*, \ldots\ldots, X_n^*$ *are generated by the moving blocks bootstrap (MBB) procedure with block size* $b(n) \to \infty$*, as* $n \to \infty$*, and that there exists* $2 < p < \infty$*,* $q > p/(p-2)$*, and* $0 < p < (p-2)/[2(p-1)]$ *such that*

(a) $\lim\limits_{k\to\infty} sup\ k^q\beta(k) < \infty$,
(b) $\mathcal{F}$ *is permissible, VC, and has envelop* F *satisfying* $P^*F^p < \infty$*, and*
(c) $\lim\limits_{n\to\infty} sup\ n^{-p}b(n) < \infty$.

Then, the MBB empirical process G_n^* converges to the same limiting process $\mathbb{H}$ as the original empirical process G_n in $\ell^\infty(\mathcal{F})$.

MBB works as follows for a stationary sample $X_1, \ldots\ldots, X_n$: For a chosen block length $b \leq n$, extend the sample by defining $X_{n+1}, \ldots\ldots, X_{n+b-1} = X_1, \ldots\ldots, X_b$, and let k be the smallest integer such that $kb \geq n$. Now, define blocks (as row vectors) $B_i = (X_i, \ldots\ldots, X_{i+b-1})$ for $i = 1, \ldots, n$ and sample from the B_is with replacement to obtain k blocks $B_1^*, \ldots\ldots, B_k^*$. The bootstrapped sample $X_1^*, \ldots\ldots, X_n^*$ consists of the first n observations from the row vector $(B_1^*, \ldots\ldots, B_k^*)$. The bootstrapped empirical measure indexed by the class $\mathcal{F}$ is then defined as $G_n^* f \equiv n^{-\frac{1}{2}} \sum_{i=1}^n \left(f(X_i^*) - P_n f\right)$.

The fast cluster bootstrap proposed in our study constructs estimators from cluster-wise sufficient statistics that are asymptotically equivalent to their full-data counterparts, while the resampling scheme fully preserves the underlying spatial dependence. Consequently, our procedure falls squarely within the theoretical framework of the moving blocks bootstrap for spatial data, satisfies all of its regularity conditions, and, therefore, inherits the conclusions of Theorem 1, guaranteeing asymptotic validity under dependence.

References

1. Lim, J.; Lee, K.; Yu, D.; Liu, H.Y.; Sherman, M. Parameter estimation in the spatial auto-logistic model with working independent subblocks. *Comput. Stat. Data Anal.* **2012**, *56*, 4421–4432. [CrossRef]
2. Wang, Y.F.; Song, Y.Q. Variable selection via penalized quasi-maximum likelihood method for spatial autoregressive model with missing response. *Spat. Stat.* **2024**, *59*, 100809. [CrossRef]
3. Al-Momani, M.; Arashi, M. Ridge-Type Pretest and Shrinkage Estimation Strategies in Spatial Error Models with an Application to a Real Data Example. *Mathematics* **2024**, *12*, 390. [CrossRef]
4. MacKinnon, J.G.; Webb, M.D. Wild bootstrap inference for wildly different cluster sizes. *J. Appl. Econom.* **2017**, *32*, 233–254. [CrossRef]
5. MacKinnon, J.G.; Webb, M.D. Pitfalls when estimating treatment effects using clustered data. *Political Methodol.* **2017**, *24*, 20–31.
6. MacKinnon, J.G.; Webb, M.D. The wild bootstrap for few (treated) clusters. *Econom. J.* **2018**, *21*, 114–135. [CrossRef]
7. Pustejovsky, J.E.; Tipton, E. Small sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *J. Bus. Econ. Stat.* **2018**, *36*, 672–683. [CrossRef]
8. Djogbenou, A.A.; MacKinnon, J.G.; Nielsen, M.Ø. Asymptotic theory and wild bootstrap inference with clustered errors. *J. Econom.* **2019**, *212*, 393–412. [CrossRef]
9. Canay, I.A.; Santos, A.; Shaikh, A. The wild bootstrap with a 'small' number of 'large' clusters. *Rev. Econ. Stat.* **2021**, *103*, 346–363. [CrossRef]
10. Hong, H.; Li, J. The numerical bootstrap. *Ann. Stat.* **2020**, *48*, 397–412. [CrossRef]
11. Taconeli, C.A.; de Lara, I.A.R. Improved confidence intervals based on ranked set sampling designs within a parametric bootstrap approach. *Comput. Stat.* **2022**, *37*, 2267–2293. [CrossRef]
12. Varmann, L.; Mouriño, H. Clustering Empirical Bootstrap Distribution Functions Parametrized by Galton–Watson Branching Processes. *Mathematics* **2024**, *12*, 2409. [CrossRef]
13. Bongiorno, C. Bootstraps regularize singular correlation matrices. *J. Comput. Appl. Math.* **2024**, *449*, 115958. [CrossRef]
14. Febrero-Bande, M.; Galeano, P.; García-Portugués, E.; González-Manteiga, W. Testing for linearity in scalar-on-function regression with responses missing at random. *Comput. Stat.* **2024**, *39*, 3405–3429. [CrossRef]
15. Freedman, D.A. Bootstrapping Regression Models. *Ann. Stat.* **1981**, *9*, 1218–1228. [CrossRef]
16. Liu, R.Y. Bootstrap procedures under some non-I.I.D. models. *Ann. Stat.* **1988**, *16*, 1696–1708. [CrossRef]
17. Mammen, E. Bootstrap and wild bootstrap for high dimensional linear models. *Ann. Stat.* **1993**, *21*, 255–285. [CrossRef]
18. Davidson, R.; Flachaire, E. The wild bootstrap, tamed at last. *J. Econom.* **2008**, *146*, 162–169. [CrossRef]
19. Cameron, A.C.; Gelbach, J.B.; Miller, D.L. Bootstrap-based improvements for inference with clustered errors. *Rev. Econ. Stat.* **2008**, *90*, 414–427. [CrossRef]
20. Flachaire, E. Bootstrapping heteroskedastic regression models: Wild bootstrap vs. pairs bootstrap. *Comput. Stat. Data Anal.* **2005**, *49*, 361–376. [CrossRef]
21. Bouzebda, S.; El-hadjali, T.; Ferfache, A.A. Central limit theorems for functional Z-estimators with functional nuisance parameters. *Commun. Stat.-Theory Methods* **2022**, *53*, 2535–2577. [CrossRef]
22. Bouzebda, S.; Elhattab, I.; Ferfache, A.A. General M-Estimator Processes and their m out of n Bootstrap with Functional Nuisance Parameters. *Methodol. Comput. Apply Probab.* **2022**, *24*, 2961–3005. [CrossRef]

23. Mackinnon, J.G. Fast cluster bootstrap methods for linear regression models. *Econom. Stat.* **2023**, *26*, 52–71. [CrossRef]
24. Rüttenauer, T. Spatial Regression Models: A Systematic Comparison of Different Model Specifications Using Monte Carlo Experiments. *Sociol. Methods Res.* **2022**, *51*, 728–759. [CrossRef]
25. Cressie, N. *Statistics for Spatial Data*; John Wiley & Sons: Nashville, TN, USA, 1993.
26. Mackinnon, J.G.; Webb, M.D. *When and How to Deal with Clustered Errors in Regression Models*; Queen's Economics Department Working Paper; Queen's University: Kingston, ON, USA, 2020.
27. Hansen, B.E. *Econometrics*; University of Wisconsin Department of Economics: Madison, WI, USA, 2020.
28. Wu, C.F.J. Jackknife bootstrap and other resampling methods in regression analysis. *Ann. Stat.* **1986**, *14*, 1261–1295. [CrossRef]
29. Roodman, D.; MacKinnon, J.G.; Nielsen, M.Ø.; Webb, M.D. Fast and wild: Bootstrap inference in Stata using boottest. *Stata J.* **2019**, *19*, 4–60. [CrossRef]
30. MacKinnon, J.G. Wild cluster bootstrap confidence intervals. *L'Actualité Économique* **2015**, *91*, 11–33. [CrossRef]
31. Mackinnon, J.G.; Nielsen, M.Ø.; Webb, M.D. Cluster-robust inference: A guide to empirical practice. *J. Econom.* **2023**, *232*, 272–299. [CrossRef]
32. Radulović, D. The bootstrap for empirical processes based on stationary observations. *Stoch. Process. Their Appl.* **1996**, *65*, 259–279. [CrossRef]