

Supplementary Material: Bayesian Estimation and Comparison of Idiographic Network Models

Björn S. Siepe¹, Matthias Kloft¹, and Daniel W. Heck¹

¹University of Marburg

Author Note

Björn S. Siepe  <https://orcid.org/0000-0002-9558-4648>

Matthias Kloft  <https://orcid.org/0000-0003-1845-6957>

Daniel W. Heck  <https://orcid.org/0000-0002-6302-9252>

Correspondence concerning this article should be addressed to Björn S. Siepe, Psychological Methods Lab, Department of Psychology, University of Marburg, Gutenbergstraße 18, Marburg, Germany.

E-mail: bjoern.siepe@uni-marburg.de

Contents

Sampling in BGGM	3
Simulation Study 1	6
Stan Implementation	20
Simulation Study 2	26
Empirical Example	28
Session Information	33

Sampling in BGGM

Let \mathbf{Y} be the matrix of observed data with n rows for k dependent variables and \mathbf{X} be the matrix of p lag-1 predictor variables (i.e. \mathbf{Y} shifted back by one time point). We deviate from the notation of Mulder and Pericchi (2018) by using \mathbf{C} instead of \mathbf{B} for the scale matrix of the matrix-F prior.

The specification of the prior follows from first using a Matrix-F prior for the precision matrix (Mulder & Pericchi, 2018):¹

$$\boldsymbol{\Theta} \sim F(\nu, \delta, \mathbf{C}), \quad (1)$$

with degrees of freedom ν , shape parameter δ , and scale matrix \mathbf{C} . Usually, however, we are not interested in the precision parameters. Instead, it is more intuitive to think about the implied partial correlations. The approach by Williams and Mulder (2020) solves this issue by allowing us to define a (marginal) prior directly on the partial correlations, specifically, a beta distribution on the interval from -1 to $+1$:

$$\rho_{ij} \sim \text{Beta}\left(\frac{\delta}{2}, \frac{\delta}{2}\right) \text{ scaled to } [-1, 1] \quad (2)$$

The standard deviation of this prior is $s_\rho = \frac{1}{\sqrt{\delta+1}}$. Hence, the user can define a prior by plugging in an expected value for the standard deviation of non-zero partial correlations by choosing δ such that $\delta = (s_\rho)^{-2} - 1$. The other prior hyperparameters ν and \mathbf{B} are fixed to specific values such that the precision matrix is approximately distributed as $IW(\delta + p - 1, \mathbf{I}_p)$, with p being the number of variables and \mathbf{I}_p a $p \times p$ identity matrix. Hence, only the prior hyperparameter s_ρ has to be set by the user. Specifying a small number for the degrees of freedom and the identity matrix as the scale matrix results in a relatively uninformative prior (see Schuurman et al., 2016).

¹ Deviating from the original notation, we use \mathbf{C} instead of \mathbf{B} to denote the scale matrix of the matrix-F prior to avoid confusion with the matrix of regression weights \mathbf{B} .

Gibbs sampling is then performed according to the following steps:

1. Obtain the scatter matrix of independent variables as the sum of the outer product of the independent variables and the prior matrix,

$$\mathbf{S}_X = \mathbf{X}^\top \mathbf{X} + \boldsymbol{\beta}_{prior},$$

where

$$\boldsymbol{\beta}_{prior} = \frac{1}{(s_\beta)^2} \cdot \mathbf{I}_k.$$

2. Initialize MCMC sampling by setting the starting value of $\boldsymbol{\Sigma}$ to the sample correlation matrix of the observed data and the starting value of $\boldsymbol{\Theta}$ to the inverse of the sample correlation matrix of the observed data (i.e. the sample precision matrix). $\boldsymbol{\Psi}$ is set to the identity matrix.
3. Draw a matrix with β -coefficients of the temporal network from a multivariate normal distribution,

$$\mathbf{B} \sim \mathcal{MVN}(\mathbf{S}_X^{-1} \mathbf{X}^\top \mathbf{Y}, \boldsymbol{\Sigma} \otimes \mathbf{S}_X^{-1})$$

4. Compute the scatter matrix of dependent variables

$$\mathbf{S}_Y = \mathbf{Y}^\top \mathbf{Y} + \mathbf{I}_k - \mathbf{B}^\top \mathbf{S}_X \mathbf{B}$$

5. Draw a new sample for the matrix $\boldsymbol{\Psi}$,

$$\boldsymbol{\Psi} | \boldsymbol{\Theta} \sim \mathcal{W}\left((\mathbf{C} + \boldsymbol{\Theta})^{-1}, \nu_{MP} + \delta_{MP} + k - 1\right)$$

where

$$\mathbf{C} = \epsilon \cdot \mathbf{I}_k$$

$$\epsilon = 0.001$$

$$\nu_{MP} = \delta + k - 1$$

$$\nu = \epsilon^{-1}$$

$$\delta_{MP} = \nu - k + 1$$

This is explained further in Williams and Mulder (2020) and Williams et al. (2020).

6. Draw a new sample for the precision matrix Θ ,

$$\Theta \mid \Psi \sim \mathcal{W}((\Psi + \mathbf{S}_Y)^{-1}, (\nu + (n - 1)))$$

7. Compute the covariance matrix Σ corresponding to the contemporaneous network,

$$\Sigma = \Theta^{-1}.$$

Simulation Study 1

In this section, we present additional analyses for the first simulation study of the paper. Specifically, we show the performance of both *BGGM* and *graphicalVAR* across a range of hyperparameters, and compare them when edges should not be thresholded. We also present an analysis of the influence of different credible interval widths on the sensitivity and specificity of thresholding with *BGGM*.

Simulation Setup

In the following, we roughly follow the simulation study reporting template by Siepe et al. (2023). We used the *doRNG*-package to set reproducible seeds across simulation conditions. Note that we use the terms ‘sample size’ and ‘number of time points’ interchangeably. In this supplement, we occasionally repeat information also included in the main manuscript such that this supplement serves as a standalone description of the simulation study.

Aims

The simulation study aimed to compare the estimation performance of LASSO GVAR with a Bayesian GVAR across different plausible data-generating mechanisms with respect to bias and correlation with true parameters. As a second aim, we aimed to investigate the specificity and sensitivity of the subset of estimation methods that also perform variable selection. As a third aim, we wanted to investigate frequentist properties of the Bayesian credible intervals.

Data-Generating Processes

Similar to previous simulation studies, sample sizes range from relatively small numbers, which are common in longitudinal studies, to very large numbers which are only found in single-case studies so far (Mansueto et al., 2023).

The first data-generating condition used a six-node network estimated on the empirical data by Fried et al. (2021) similar to Mansueto et al. (2023). This data-generating

process is called *Empirical Sparse* hereafter because 24/36 and 6/15 edges were originally estimated to be zero for the temporal and contemporaneous network, respectively. Mean absolute non-zero edge sizes were 0.123 (temporal) and 0.165 (contemporaneous).

As a second data-generating process, we created a six-node and eight-node sparse *Simulated Chain Graph* following Hoekstra et al. (2022), because previous simulation studies have shown good performance of `graphicalVAR` with such data structures (Hoekstra et al., 2022). In a chain graph, each node is only connected to two neighboring nodes in both networks. Mean absolute non-zero edge sizes were 0.333 (temporal) and 0.358 (contemporaneous) for the six-node graph and 0.343 for both networks of the eight-node graph.

Third, we created a non-sparse data-generating process by adding random numbers drawn from a uniform distribution with the maximum set to the largest coefficient in the estimated network to all zero-coefficients of the *Empirical Sparse* graph. This graph is referred to as *Simulated Nonsparse* throughout the manuscript. We added this last condition to investigate performance when not assuming a sparse network as ground truth, similar to previous investigations in the cross-sectional network literature (Epskamp et al., 2017) while keeping ourselves to realistic parameter sizes. Mean absolute non-zero edge sizes were 0.145 (temporal) and 0.142 (contemporaneous).

Estimands/Targets

The targets of our simulation study were the temporal and contemporaneous parameters of the networks.

Methods

Network Estimation with graphicalVAR

Based on previous work establishing the multivariate regression with covariance estimation (MRCE) algorithm (Rothman et al., 2010), `graphicalVAR` jointly estimates the temporal and contemporaneous coefficients using cyclical-coordinate descent (Epskamp, Waldorp, et al., 2018). Separate LASSO regularization parameters λ_B and λ_Θ are placed on

the temporal coefficient matrix \mathbf{B} and the precision matrix Θ , respectively.² The GVAR model is estimated and compared based on the Extended Bayesian Information Criterion (EBIC; Chen & Chen, 2008). The sparsity of the solution is controlled by the EBIC hyperparameter γ , where higher values lead to more sparse networks. To increase the sensitivity for detecting nonzero edges in practical applications, γ is often set to 0 which reduces the EBIC to the regular BIC (Mansueto et al., 2023). Here, we set γ to either 0 or 0.5. We estimated regularized graphicalVAR models using the default grid of 50×50 penalty parameters for the temporal and contemporaneous coefficients.

Network Estimation with BGGM

We used a grid of priors ranging from informative to rather diffuse priors. Specifically, we set the standard deviation of the prior distribution on the coefficients of the temporal network to $s_\beta \in \{0.2, 0.5, 1.0\}$ and those of the contemporaneous network to $s_\rho \in \{0.1, 0.3, 0.5\}$. We then created a grid of all possible prior combinations, resulting in nine different prior configurations. We used 90%, 95%, and 99% credible intervals and fitted the model using a Gibbs sampler with 50 burn-in and 50,000 sampling iterations.

The large number of Gibbs sampler iterations was chosen because the computational costs of the sampler were negligible and because the autocorrelation across iterations for the contemporaneous network was relatively high. For example, drawing 50,000 MCMC samples for a model with six nodes, 200 time points, and default priors required only about four seconds on an Intel i7-1260P processor. Convergence was checked visually via autocorrelations, trace plots, and effective sample sizes. More information on convergence statistics is given in the section on the empirical example below.

Performance Measures

We define the absolute bias per model matrix as the mean of the absolute differences between the estimated and the true coefficient matrix. We then calculate the estimated absolute bias per condition by calculating the mean of the absolute biases across simulation

² In the notation of Epskamp, van Borkulo, et al. (2018), our Θ corresponds to their \mathbf{K} .

repetitions.

We define the correlation with the true parameters per model matrix as the Pearson correlation between the estimated and the true coefficient matrix. We then calculate the estimated correlation per condition by calculating the mean of absolute biases across simulation repetitions.

We define sensitivity per model matrix as the sum of true positive edges divided by the sum of the sum of true positive edges and the sum of false negative edges. We define specificity per model matrix as the sum of true negative edges divided by the sum of the sum of true negative edges and the sum of false positive edges. In both cases, the mean performance measure is then again obtained by averaging over simulation repetitions.

Monte Carlo standard errors were calculated using the formulas by Morris et al. (2019) and Siepe et al. (2023). Standard errors for correlations were calculated as

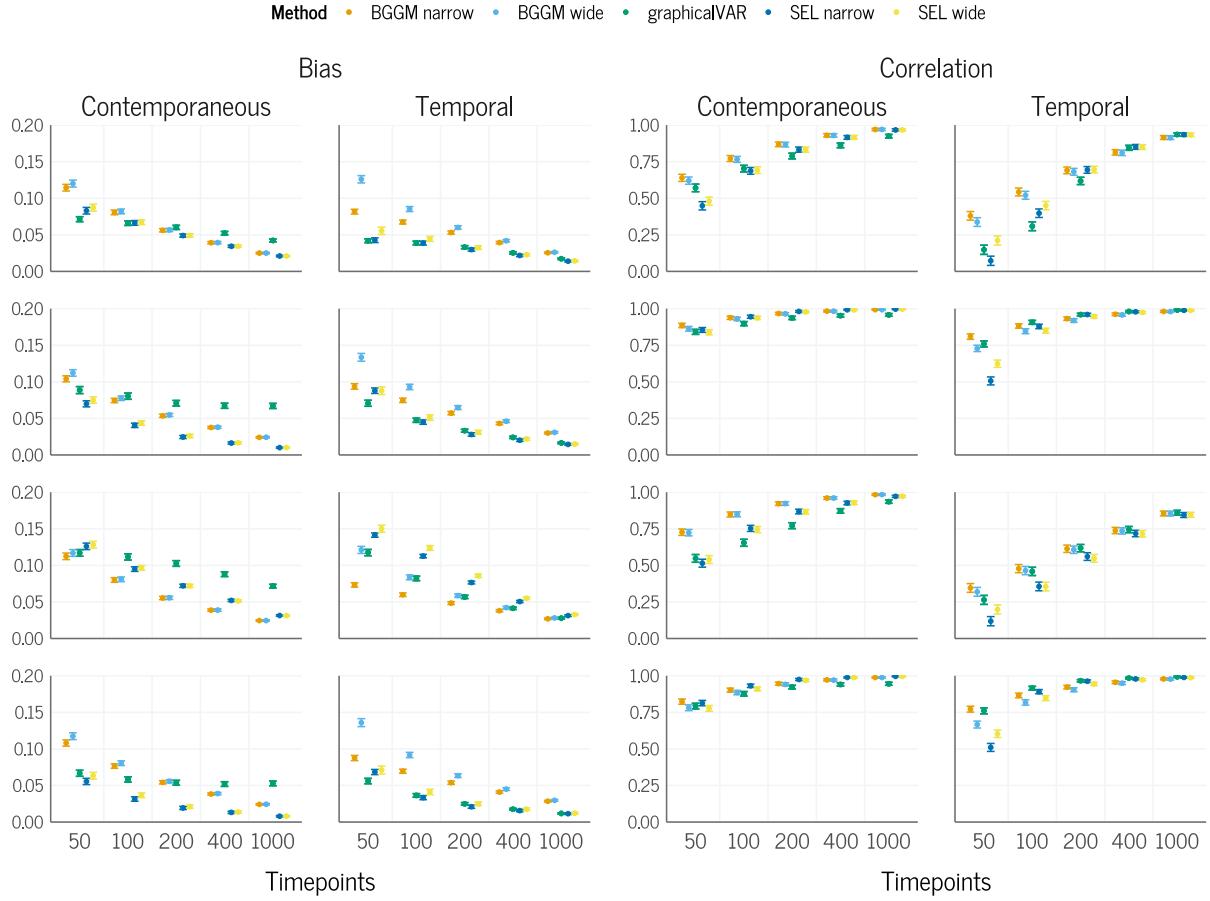
$SE = \sqrt{1 - r^2} / \sqrt{n_{rep} - 2}$, where r was the estimated correlation for a condition and n_{rep} was the number of simulation repetitions.

Main Results

In this section, we present the main results of our simulation study. We present some additional results which we deem slightly less central in the section “Additional Results” below. Figure 1 shows the performance of Bayesian estimation for different priors with (denoted as “SEL”) and without (denoted as “BGGM”) thresholding (based on 95%-CIs). Moreover, the plot shows the results for LASSO estimation with the EBIC hyperparameter γ set to 0. Overall, the absolute bias of the different methods was comparable in many conditions, with the largest differences occurring for small sample sizes. As expected, all estimation methods performed better and showed increasingly similar performance with more time points per individual. However, LASSO estimation showed a relatively smaller performance gain from more time points compared to other methods. At a small sample size of 50 time points, the overall performance of all methods was mediocre at best, with correlations falling below .50 for the temporal networks of the empirical sparse and simulated

Figure 1

Absolute Bias and Correlation with True Edges for Different Simulation Conditions.



Note. Separated by contemporaneous and temporal network (columns) and the data-generating processes (rows). Estimation methods are shown in different colors in the same order as they appear in the legend. “BGGM” denotes Bayesian estimation without thresholding, “SEL” denotes Bayesian estimation with thresholding. Vertical bars indicate $\pm 1 \times SE$.

nonsparse graphs.

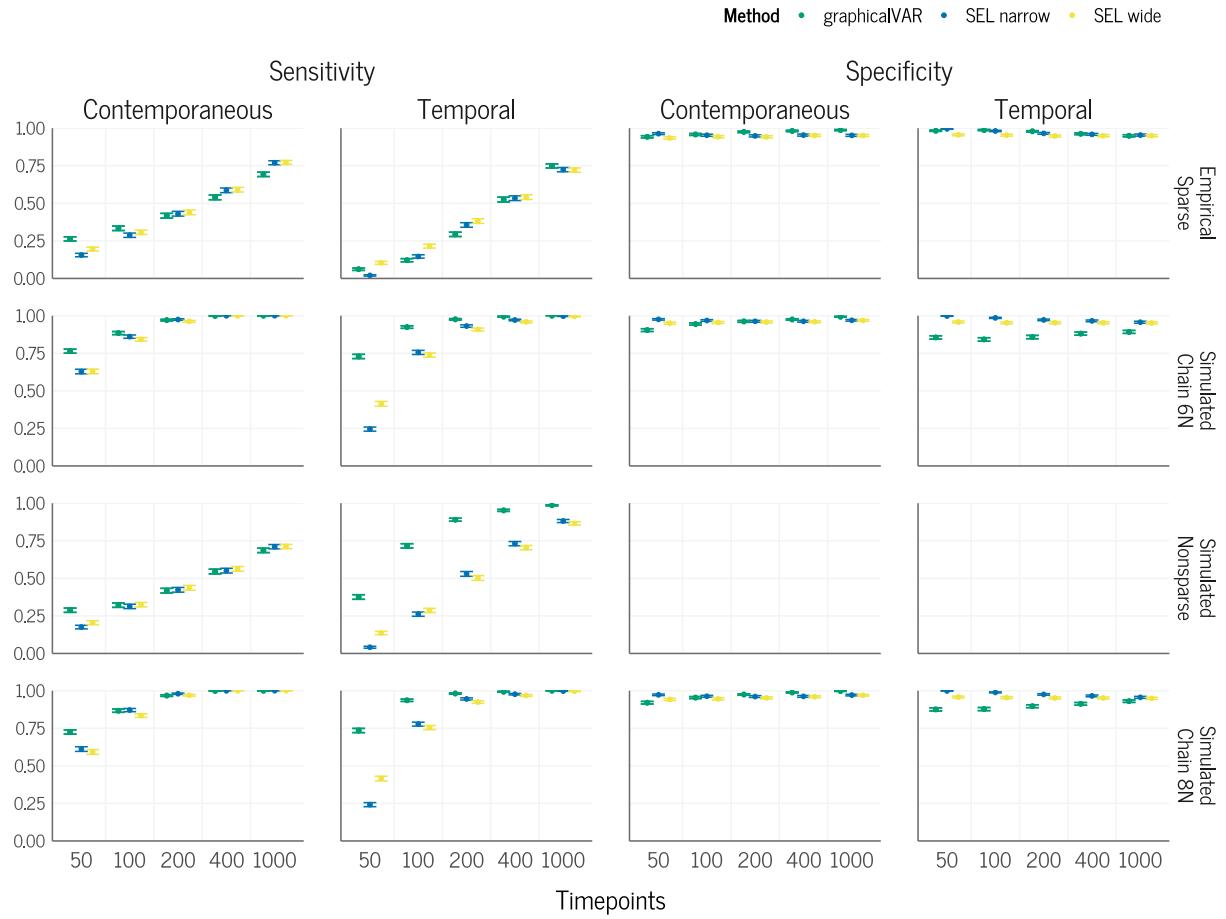
Which of the methods performed best depended on the data-generating process and the performance measure. In terms of bias, LASSO and thresholded Bayesian estimates with narrower priors performed best and showed very similar performance for sparse graphs, while non-thresholded Bayesian GVAR performed worse in most conditions with smaller sample sizes. In sparse graphs, there was no clear winner between thresholded Bayesian estimates and LASSO in terms of bias. However, in the non-sparse graph, non-thresholded Bayesian estimation with narrower priors performed best with regard to both bias and correlation. As

the comparison of non-thresholded Bayesian estimation and LASSO regularization may be considered unfair when the data-generating process is dense, we also show results of comparing the former method with nonregularized **graphicalVAR** in the Section Additional Results below. The results show that Bayesian GVAR performed as well or better for all data-generating processes, especially when inducing stronger regularization via a slightly narrower prior.

All methods performed better for contemporaneous networks than for temporal networks in terms of correlation, but not necessarily for bias, which is in line with previous simulations on the performance of LASSO GVAR (Mansueto et al., 2023). The difference was particularly striking in the nonsparse graph, where correlations were about twice as high in the contemporaneous as in the temporal network. However, there are more coefficients in the temporal than in the contemporaneous network, and thus, estimation may generally be more difficult.

For correlations, the thresholded Bayesian methods generally performed worse than other methods, especially for smaller sample sizes. With few time points, the non-thresholded Bayesian GVAR had a higher correlation with true parameters than other methods, probably due to the high degree of sparsity of both LASSO and thresholding when the sample size is small. The narrower and thus more informative prior performed as well as or even better than the more diffuse prior for both the non-thresholded and thresholded methods with regard to all performance criteria. When investigating other prior choices for our data, we generally found that a narrower prior on β and a relatively wider prior on ρ seemed to perform best. These results are presented only in the additional results section below since we chose the specific hyperparameters for the ‘wide’ and the ‘narrow’ prior before analyzing the simulation results. Thereby, we avoid cherry-picking of an optimal prior setup for BGGM after seeing the results.

Figure 2 shows the specificity and sensitivity of LASSO estimation and Bayesian estimation with thresholding based on 95% CIs. Overall, Bayesian thresholding was

Figure 2*Sensitivity and Specificity for Bayesian Thresholding and LASSO Estimation.*

Note. Estimation methods are shown in different colors in the same order as they appear in the legend. “SEL” denotes Bayesian estimation with thresholding. Specificity for *Simulated Nonsparse* is not shown, as there are no true-zero edges. Vertical bars indicate $\pm 1 \times SE$.

conservative, leading to poor sensitivity below .50 in smaller sample sizes, especially in the temporal network. A sensitivity below .50 means that less than half of the true edges were detected by Bayesian thresholding, which partly explains its poor performance in terms of the correlation with true parameters reported above. For the temporal networks, LASSO estimation generally performed better. Regarding the contemporaneous networks, LASSO estimation also had a slightly better sensitivity for sample sizes up to 100 but became similar to Bayesian estimation for larger sample sizes. Conversely, Bayesian thresholding had a slightly higher specificity for most network structures in sample sizes up to 100. This was

particularly pronounced in the temporal network of the chain graphs.

Although not necessarily a core objective of Bayesian inference, we further explored the coverage of credible intervals. Overall, coverage was good and close to the nominal values of 90%, 95%, and 99%, with a median absolute difference of 0.6% between empirical coverage and credible interval width across all simulation conditions. With more time points, coverage decreased slightly for the temporal network and increased slightly for the contemporaneous network. The choice of prior distribution affected coverage only to a small degree while the effect was larger for smaller credible intervals. The additional results below contain plots of coverage and the width of credible intervals, as well as an assessment of prior sensitivity and different settings for `graphicalVAR`.

Summary

Overall, we found that LASSO performs well for estimating idiographic networks based on longitudinal data, especially under sparse data-generating processes. Bayesian GVAR also shows a good performance which was often comparable to or, depending on the data, sometimes even higher than that of LASSO. The results suggest that in sparse graphs, both thresholding and LASSO outperformed non-thresholding methods, with LASSO and thresholded Bayesian estimation with narrower priors performing roughly equally well in terms of bias. For a non-sparse graph, continuous regularization via priors outperformed thresholding and LASSO methods in terms of bias and correlation. Overall, the results show that the match between the structure of the data-generating process and the type of estimation method determines performance. If true networks are dense, which is a plausible assumption given that edges may often not be perfectly zero, the regularization by LASSO may not always be the best option available.

In all conditions, a narrower, but not overly informative prior worked better than a wider, diffuse prior. Returning to the illustration of different priors in Figure 1 of the main manuscript, this result may not be surprising, as the wide prior places a considerable amount of prior probability on relatively large values that are rather implausible in temporal

networks with standardized coefficients. We therefore recommend that researchers first examine the prior distribution for different sets of hyperparameters (similar as in Figure 1 of the manuscript) to choose a suitable, possibly narrower prior than the default implemented in BGGM.

With respect to the frequentist properties of CIs, Bayesian estimates showed good coverage, suggesting that the posterior distribution provides a good indicator of estimation uncertainty. However, a major drawback of Bayesian estimation of GVAR models seems to be the potentially low sensitivity to detect non-zero edges in small sample sizes when using thresholding based on credible intervals. A 95% credible interval was often too wide, and hence, many edges that are nonzero in the population were set to zero. This may be due to the fact that the thresholding approach was not specifically designed for the task of reliable edge detection or structure selection (Sekulovski et al., 2023). Instead, thresholding based on credible intervals is a pragmatic ad-hoc solution as it simply dichotomizes continuous posterior distributions of edges. Still, sensitivity can be increased by using smaller credible intervals (e.g., 80% CIs) with the disadvantage of achieving lower specificity. A plot that illustrates this trade-off can be found in the additional results below. In general, our simulation provides limited information about sensitivity and specificity under various conditions, since we only focused on a limited number of hyperparameters for both Bayesian and LASSO GVAR.

We have shown that Bayesian estimation provides a viable alternative for estimating idiographic networks. Besides providing good estimation performance, Bayesian inference has the advantage of providing samples from the posterior distribution for all parameters.

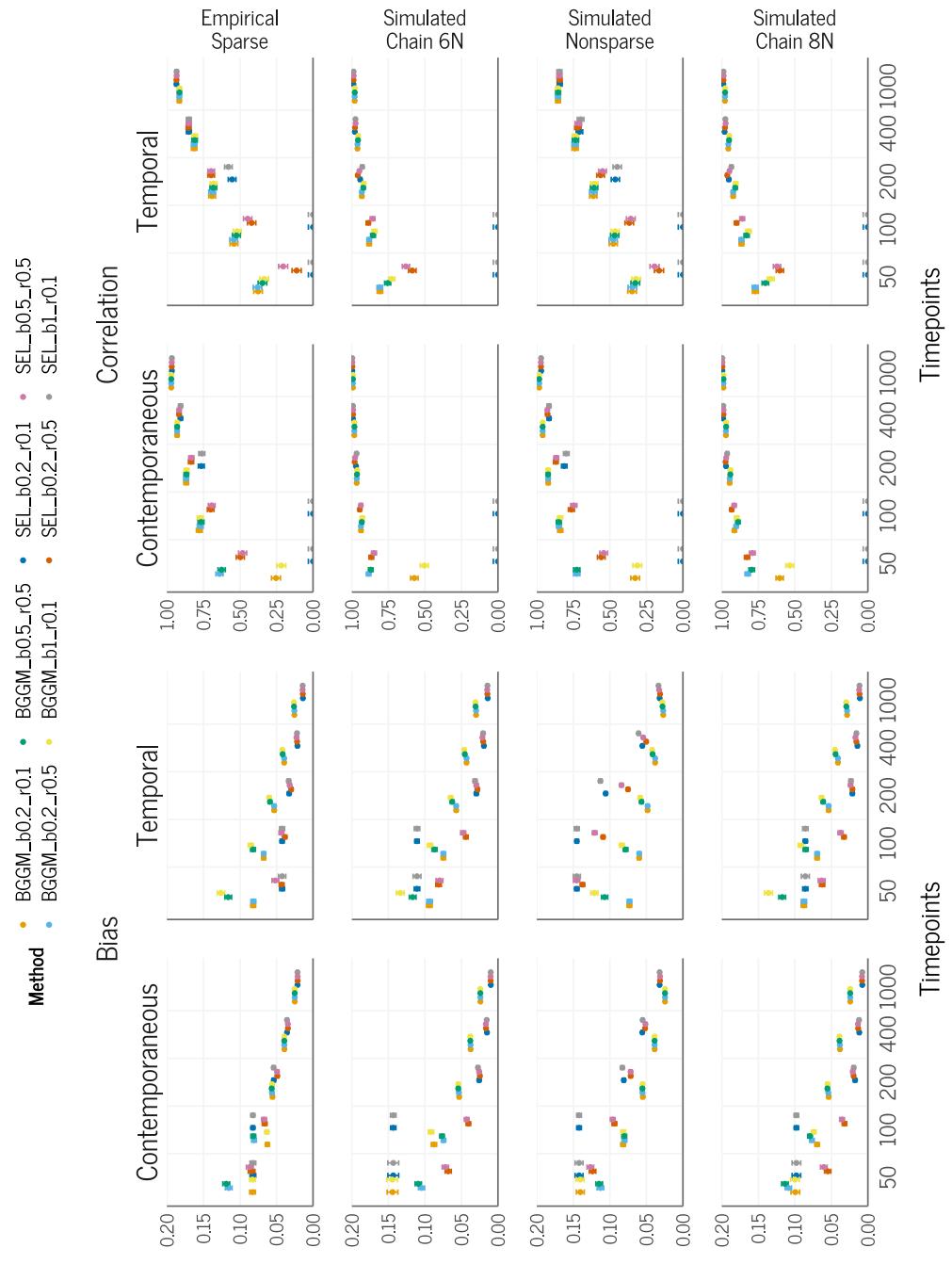
Additional Results

BGGM Evaluation

In Figure 3, we show bias and correlation with true edges of Bayesian gVAR for various prior hyperparameters. In general, narrower priors seem to perform better than wider priors. However, thresholding with a narrow prior of $s_\beta = 0.2$ and $s_\rho = 0.1$ or $s_\beta = 1$

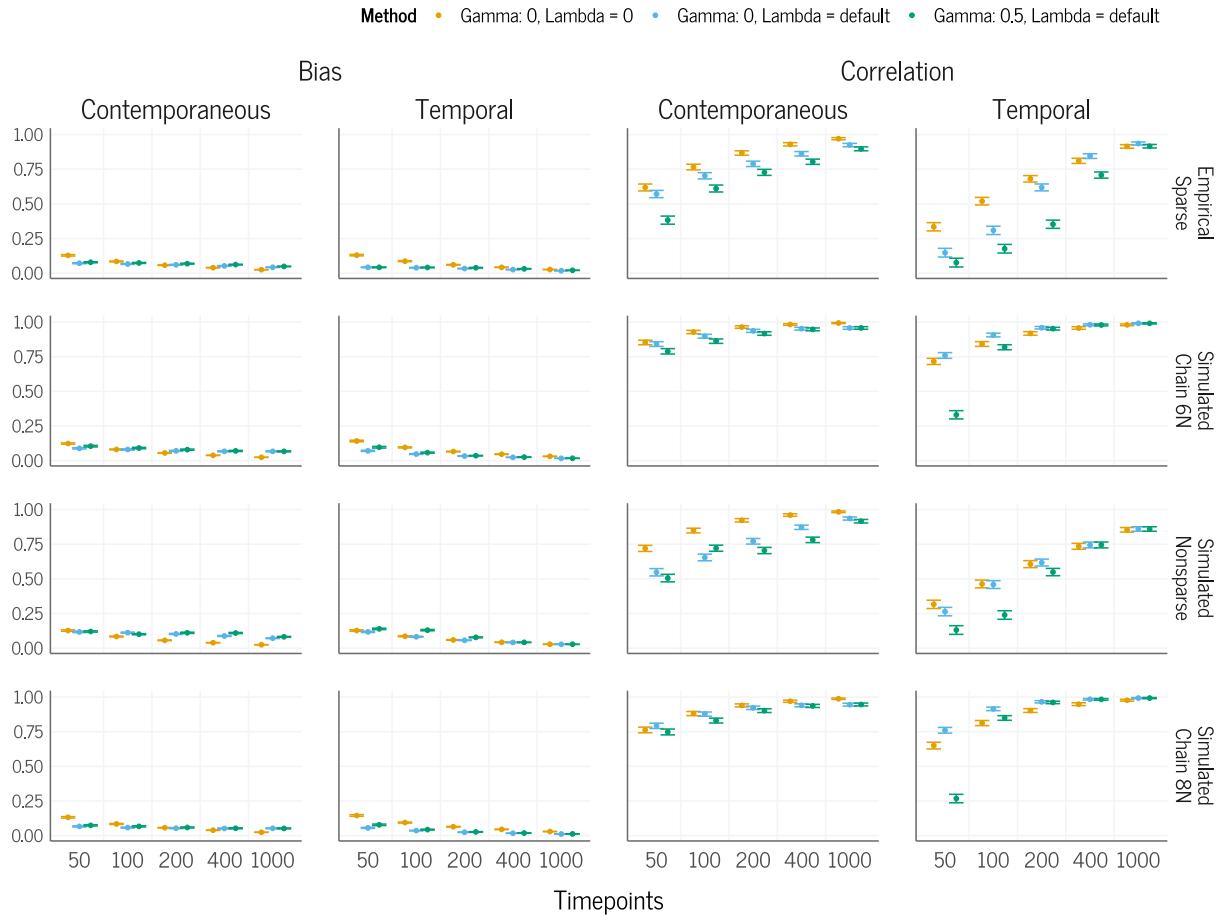
and $s_\rho = 0.1$ does not work well, especially when using correlations with true values as performance metric. Likely, resulting networks are mostly empty due to estimates being pushed towards zero, especially with a narrow prior on the contemporaneous network. In the sparse data-generating processes, thresholding with $s_\beta = 0.2$ and $s_\rho = 0.5$ performed best with regards to bias, while non-thresholding performed better in the nonsparse data-generating process. For correlations with true parameters, non-thresholded estimation with $s_\beta = 0.2$ and $s_\rho = 0.5$ generally seemed to work best, although differences with other hyperparameters may be due to sampling variability as indicated by the error bars.

Figure 3
Performance of BGGM with Different Priors.



Note. Bias and correlation with true edges for different simulation conditions for the contemporaneous and temporal network (columns) and the data-generating processes (rows) for different priors of BGGM. ‘BGGM’ indicates non-thresholding, ‘SEL’ indicates thresholding. The value after ‘b’ indicates the prior standard deviation s_β for the temporal network. The value after ‘r’ indicates the prior standard deviation s_ρ for the contemporaneous network. Vertical bars indicate $\pm 1 \times SE$.

Figure 4
Performance of graphicalVAR with different settings.



Note. Bias and correlation with true edges for different simulation conditions for the contemporaneous and temporal network (columns) and the data-generating processes (rows) for different settings of graphicalVAR. γ denotes the EBIC hyperparameter, λ denotes the choice of regularization parameters. Vertical bars indicate $\pm 1 \times SE$.

graphicalVAR Evaluation

In Figure 4, we show bias and correlation with true edges for LASSO gVAR for different settings of graphicalVAR.

Specifically, we varied the choice of the EBIC hyperparameter γ and using regularization, indicated by using the default values for λ , or not performing regularization. Regarding bias, regularized estimation performs better in small sample sizes, whereas non-regularized estimation performs equally well or better in large sample sizes. The picture

is not so clear for correlation with true parameters, where nonregularized estimation is often slightly or substantially better at smaller sample sizes for three out of four of the contemporaneous networks and two out of four of the temporal networks. Correlations with true parameters tend to be relatively low for regularized estimation in small sample sizes, which can likely be explained by many coefficients being set to zero by LASSO.

Non-Thresholded Evaluation

In Figure 5, we compare non-thresholded estimation with no edges being set to zero in the Bayesian gVAR framework in `BGGM` with non-regularized estimation in `graphicalVAR`. We see that a narrow prior in `BGGM` performs as good or better as unregularized estimation in `graphicalVAR` and the wider prior in `BGGM`.

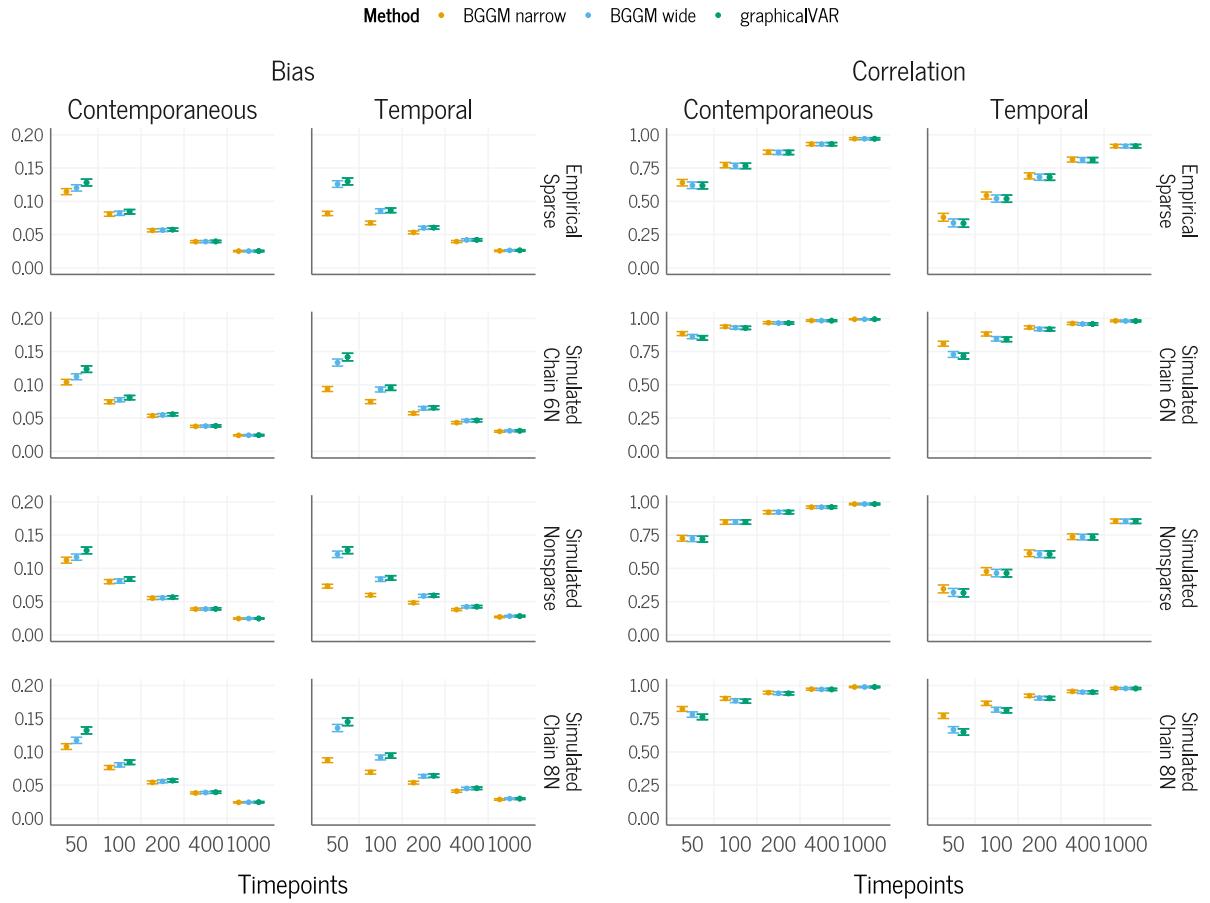
Different Credible Intervals

In Figure 6, we show specificity and sensitivity of the Bayesian gVAR with different credible interval (CI) widths used for thresholding.

As expected, sensitivity is considerably higher when using a more narrow CI width of .90 compared to .99, which is especially pronounced in smaller sample sizes. As a trade-off, specificity is smaller for the narrower CIs. Notably, this does not seem to be the case for the temporal network with smaller sample sizes, where specificity is comparable across all CI widths.

Figure 5

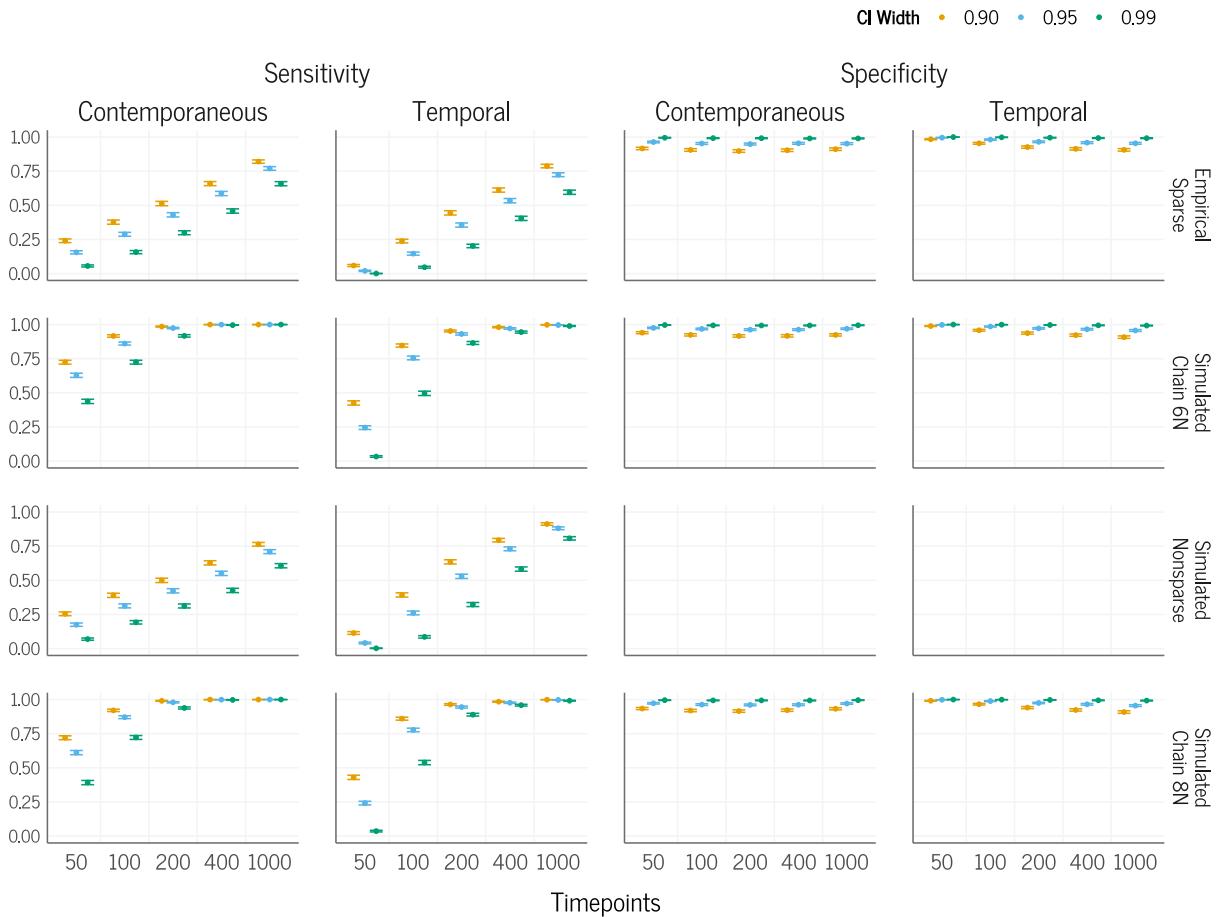
Comparison of BGGM and graphicalVAR without thresholding.



Note. Bias and correlation with true edges for different simulation conditions for the contemporaneous and temporal network (columns) and the data-generating processes (rows). Different estimation methods shown in different colors. Both Bayesian and LASSO gVAR without thresholding. Vertical bars indicate $\pm 1 \times SE$. The narrow prior has hyperparameters $s_\rho = 0.3$ and $s_\beta = 0.2$, the wide prior has hyperparameters $s_\beta = 1$ and $s_\rho = 0.5$.

Figure 6

Performance of BGGM with Different Credible Intervals.



Note. Sensitivity and Specificity for different interval widths for BGGM gVAR using a narrow prior. Specificity not defined for the simulated nonsparse graph, as all edges are nonzero.

Stan Implementation

In this section, we explain our Stan implementation of the Bayesian GVAR model in Stan. We further present simulation results on its performance compared to BGGM.

Aims

The goal of this simulation was to compare the Bayesian implementations of Bayesian GVAR models with respect to their parameter recovery performance. A secondary aim of this simulation was to compare the differences between the point estimates of the Bayesian modeling approaches. The overarching intention of this simulation study was not to

exhaustively show possible performance differences between implementations, but rather to show that their performance is similar enough that they can be used interchangeably in common applications. Therefore, we restricted both the design and the number of repetitions to keep computational efforts reasonably low.

Data-Generating Mechanisms

We generated data based on the data-generating mechanisms of the empirical sparse network, simulated sparse network, and simulated non-sparse network in the main study, to which we refer for more information. We generated data under 3 (sample size) \times 3 (data-generating processes) = 9 simulation conditions. We used three different sample sizes $n \in \{50, 200, 400\}$ that range from a lower sample size to a realistic upper bound of observations found in most psychological experience sampling studies.

Estimands

The targets of our simulation study were the temporal and contemporaneous parameters of the networks and the effective sample size of the sampling algorithms.

Methods

We compared three different implementations of a Bayesian GVAR: First, we used the `BGGM` package that uses Gibbs sampling to obtain posterior distributions of all parameters as described in the manuscript. Here, we used a $s_\beta = 0.5$ prior for β and $s_\rho = 0.25$ for ρ . We used 50,000 Monte Carlo iterations and 500 burn-in iterations, resulting in 49,500 samples per model.

Second, for the implementation in Stan, we chose the same general model, but with either a Lewandowski-Kurowicka-Joe prior (LKJ, Lewandowski et al., 2009) or an Inverse-Wishart prior (see, e.g., Schuurman et al., 2016) for the innovation covariance matrix. Both of these methods are implemented within the `stan_gvar` function in the `tsnet` package. As the LKJ implementation does not allow for direct specifications of priors on the partial correlations, we implemented a workaround to enable priors on specific partial correlations. We consider this feature experimental and would advise users wishing to implement

edge-specific priors in the contemporaneous network to preferentially use IW priors. For the temporal regression parameters of \mathbf{B} , we assigned the same weakly informative priors in both versions of the model:

$$\beta_{ij} \stackrel{i.i.d.}{\sim} \mathcal{N}(0, 0.5). \quad (3)$$

The standard LKJ implementation of the model does not allow for edge-specific priors on the partial correlations, which is possible with an inverse Wishart prior. To mimic this, we use the workaround of a “joint” prior that, in addition to the LKJ prior on the matrix itself, enables the specification of additional distinct beta priors for each of the partial correlations.³ Therefore, we first assigned an uninformative LKJ prior directly to the Cholesky factor decomposition (Barnard et al., 2000) of the correlation matrix of innovations:

$$\boldsymbol{\Omega}_L \sim \text{LKJ-Cholesky}(\eta), \quad (4)$$

For $\eta = 1$, this prior choice implies a symmetric marginal scaled beta distribution on the zero-order correlations ω_{ij} (Lewandowski et al., 2009):

$$\frac{\omega_{ij} + 1}{2} \sim \text{Beta}\left(\frac{p}{2}, \frac{p}{2}\right) \quad \text{for } i < j. \quad (5)$$

By pre- and post-multiplying with standard deviations, we computed the covariance matrix Σ and consecutively its inverse, the precision matrix Θ .

$$\begin{aligned} \Sigma &= (\text{diag}(\boldsymbol{\sigma}) \boldsymbol{\Omega}_L \text{diag}(\boldsymbol{\sigma}) \boldsymbol{\Omega}_L')', \\ \log(\boldsymbol{\sigma}) &\sim t_3(0, 2), \\ \Theta &= \Sigma^{-1}. \end{aligned} \quad (6)$$

From the off-diagonal elements of the precision matrix Θ , we then computed the partial

³ This approach was inspired by a blog post by Stephen R. Martin (<http://srmartin.in/informative-priors-for-correlation-matrices-an-easy-approach/>).

correlations:

$$\rho_{ij} = \frac{-\Theta_{ij}}{\sqrt{\Theta_{ii}\Theta_{jj}}} \quad \text{for } i < j. \quad (7)$$

The second part of the mixed prior is a beta prior on each of these partial correlations. This prior was assigned by transforming the partial correlations to the interval of $[0, 1]$ and then assigning a proportional (mean-variance parameterized) beta prior:

$$\frac{\rho_{ij} + 1}{2} \sim \text{Beta}_{\text{prop}}(0.5, \sqrt{.5}) \quad \text{for } i < j. \quad (8)$$

A beta location parameter of 0.5 thus translates to an expected correlation of zero. The variance parameter of $\sqrt{.5}$ implies a uniform distribution of partial correlations. For the simulation study, we used the same beta prior for all partial correlations. In `tsnet`, users can provide a matrix for each location and scale of the proportional beta distribution to set edge-specific priors, where the additional beta distribution provides edge-specific regularization. In the fitting functions of the `tsnet` package, the user can specify the LKJ parameter in terms of the implied marginal beta distribution of zero-order correlations on the interval $[-1, 1]$. The edge-specific priors need to be specified in the proportional Beta metric. We consider this to be an experimental feature.

In the second version of the model we assigned an Inverse-Wishart prior directly to the innovation covariance matrix:

$$\begin{aligned} \Theta &\sim \mathcal{IW}(\delta + p - 1, \mathbf{I}_p), \\ \delta &= s_\rho^{-1} - 1 \quad \text{with} \quad s_\rho = 0.25, \end{aligned} \quad (9)$$

where p is the number of variables in the network, \mathbf{I}_p is an identity matrix of order p , and $s_\rho = 0.25$ is the standard deviation of the implied marginal beta distribution of the partial correlations. From the precision matrix, we then computed the covariance matrix:

$$\Sigma = \Theta^{-1}. \quad (10)$$

In the `tsnet` package, users can specify a scaling matrix instead of \mathbf{I}_p to further assume edge-specific priors on partial correlations.

In both versions (LKJ and IW) of the model, the likelihood for the variables at a single time point is then given by (assuming the same notation as in the manuscript):

$$\mathbf{y}_t = \mathbf{B}\mathbf{y}_{t-1} + \boldsymbol{\zeta}_t \quad (11)$$

$$\boldsymbol{\zeta}_t \sim \mathcal{N}(\mathbf{0}, \Sigma). \quad (12)$$

For both Stan implementations of the model, we used the `rstan` R package (Stan Development Team, 2024) to run four Hamiltonian Monte Carlo (HMC) chains of the Stan No U-Turn (NUTS) sampler for 500 iterations (plus 500 burn-in iterations), resulting in 2,000 samples per model fit. We used a target average acceptance probability (`adapt_delta`) of .8.

Performance Measures

We used bias and mean squared error (MSE) as defined in the other simulation studies and in Siepe et al. (2023). We used the formulas in Siepe et al. (2023) to compute the MCSE for the performance measures, where we treated the mean absolute difference as a generic statistic. We additionally computed the effective sample size (ESS) of all methods using the `coda` R package (Plummer et al., 2020).

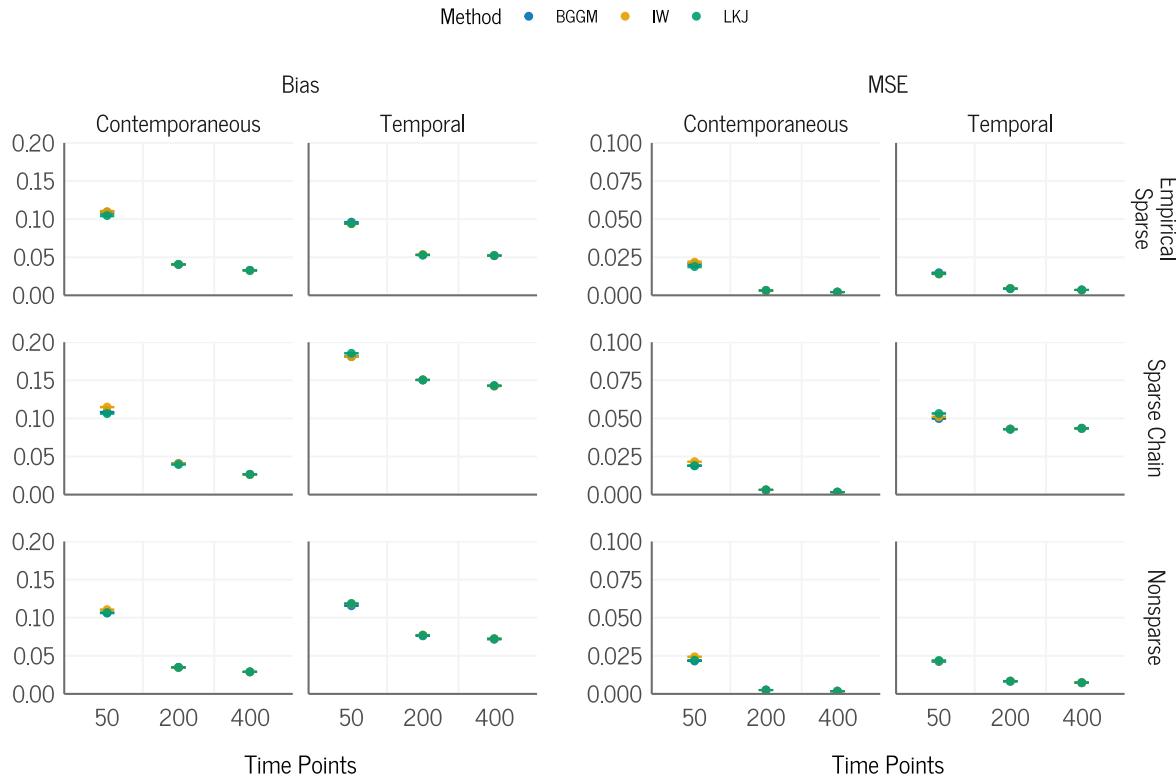
Results

The results of our simulation concerning bias and MSE are shown in Figure 7. As expected, the performance of all implementations increases with a larger number of time points. The performance of all implementations is virtually identical for all simulated conditions.

Convergence

In Figure 8, we show the ESS of the different Bayesian GVAR implementations. As we used a different number of iterations for BGGM (50,000) than for both Stan implementations (2,000), we computed the relative ESS by dividing the ESS by the number

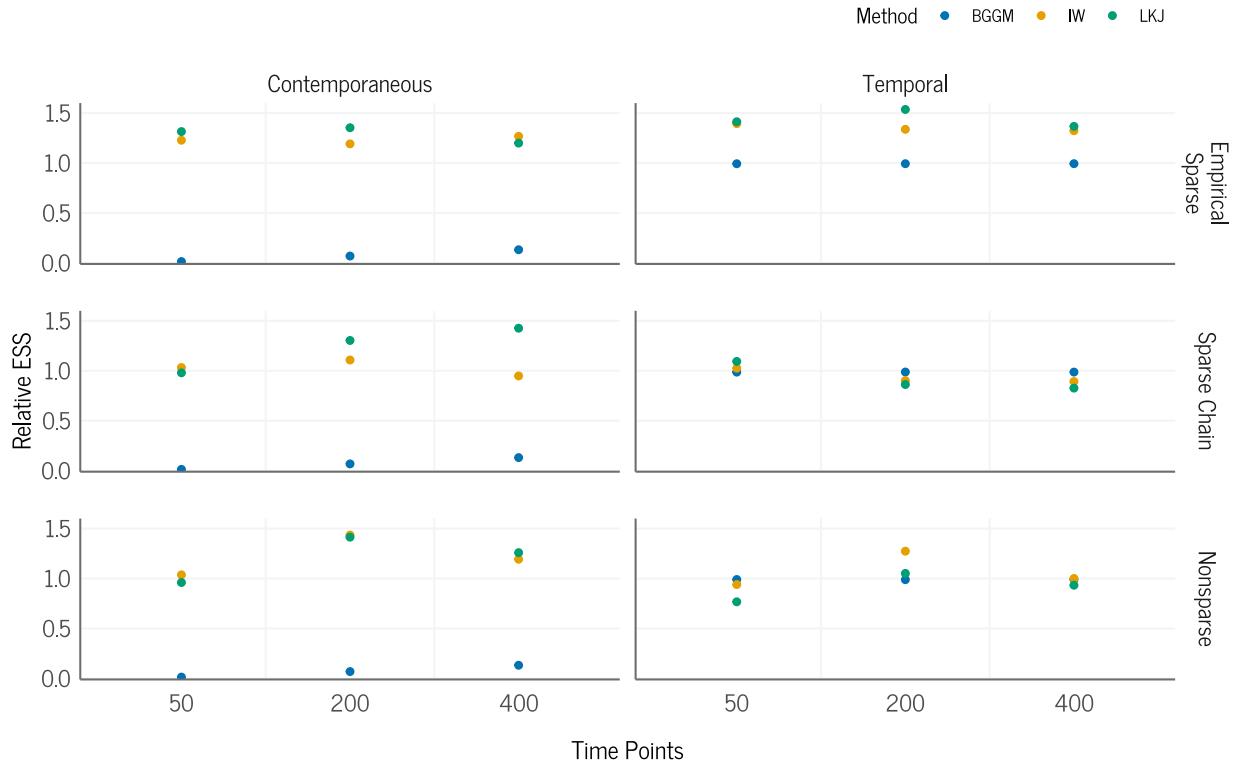
Figure 7
Simulation Results Bayesian GVAR Implementation



Note. Absolute bias and MSE separated by contemporaneous and temporal network (columns) and the data-generating processes (rows). Estimation methods are shown in different colors in the same order as they appear in the legend. Vertical bars indicate $\pm 1 \times SE$.

of sampling iterations. Both Stan implementations are more efficient than BGGM for the contemporaneous networks studied here, in so far as they have a relatively high ESS compared to the overall number of iterations. This arises due to the high autocorrelation of the BGGM sampler for the contemporaneous network. For the temporal network, there is no such clear trend.

Figure 8
Effective Sample Size Bayesian GVAR Implementation



Note. Effective sample size separated by contemporaneous and temporal network (columns) and the data-generating processes (rows). Estimation methods are shown in different colors in the same order as they appear in the legend.

Simulation Study 2

In this section, we present additional analyses for the second simulation study. We show an example of the change matrices that were used to manipulate the data-generating processes to create differences of various strengths. Further, we show the distribution of test values under the Null.

Change Matrices

We did not use pre-determined change matrices, as we occasionally ran into issues with non-semi-positive definite precision matrices. Rather, we iteratively drew change matrices, added them to the precision matrix, and then checked for semi-positive definiteness.

The approach is described in detail in the R code on OSF (<https://osf.io/9byaj/>).

To illustrate this approach, we show the change matrix for the ‘constant 0.15’ condition for the simulated nonsparse graph. First, we see the change matrix that was added to the temporal network:

$$\begin{bmatrix} 0.15 & -0.15 & -0.15 & -0.15 & -0.15 & 0.15 \\ -0.15 & -0.15 & 0.15 & -0.15 & -0.15 & 0.15 \\ 0.15 & -0.15 & -0.15 & -0.15 & -0.15 & -0.15 \\ 0.15 & -0.15 & 0.15 & -0.15 & -0.15 & -0.15 \\ -0.15 & -0.15 & 0.15 & 0.15 & 0.15 & -0.15 \\ -0.15 & 0.15 & 0.15 & -0.15 & 0.15 & -0.15 \end{bmatrix}$$

Second, we see the change matrix that was added to the precision matrix Θ :

$$\begin{bmatrix} 0 & 0.233 & -0.157 & 0.155 & 0.152 & -0.152 \\ 0.233 & 0 & -0.136 & 0.15 & 0.152 & 0.15 \\ -0.157 & -0.136 & 0 & 0.151 & 0.152 & -0.149 \\ 0.155 & 0.15 & 0.151 & 0 & 0.149 & 0.108 \\ 0.152 & 0.152 & 0.152 & 0.149 & 0 & 0.138 \\ -0.152 & 0.15 & -0.149 & 0.108 & 0.138 & 0 \end{bmatrix}$$

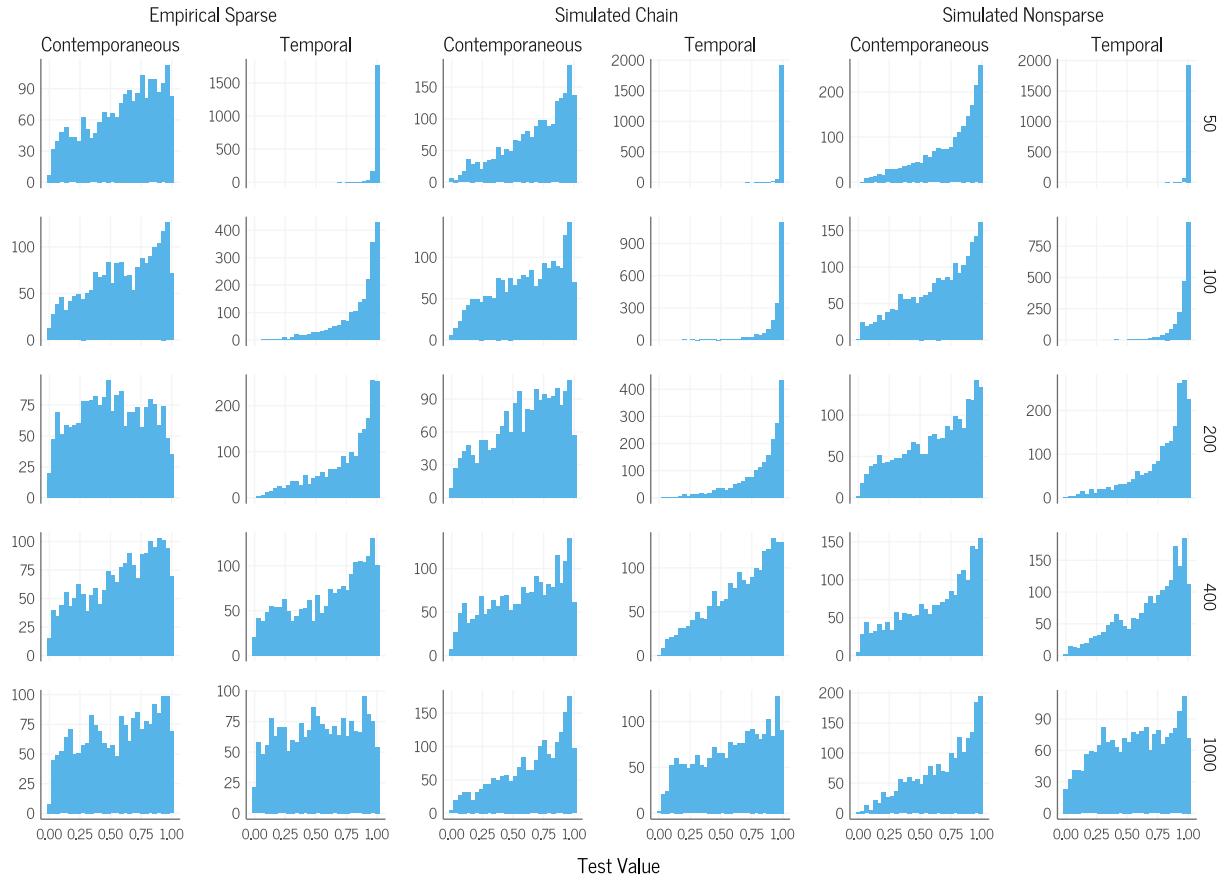
It is scaled to the diagonal elements of the precision matrix to achieve the desired change in the partial correlations.

Distribution under the Null

The following plots show the distribution of test values under the null when using the Frobenius norm, with the narrow prior of $s_\rho = 0.3$ and $s_\beta = 0.2$ in Figure 9 the wide prior $s_\beta = 1$ and $s_\rho = 0.5$ in Figure 10. For the narrow prior, test values were left-skewed, especially for the temporal network. This can likely be explained by the fact that a narrow prior draws all estimates towards zero, making networks more similar across different data-generating processes. This tendency is therefore less pronounced for the wider prior, where sampling distributions of the test value are more uniform.

Figure 9

Distribution of Test Values under the Null for the Narrow Prior.



Note. Using Frobenius norm only. X-axis displays different data-generating processes, y-axis displays different sample sizes.

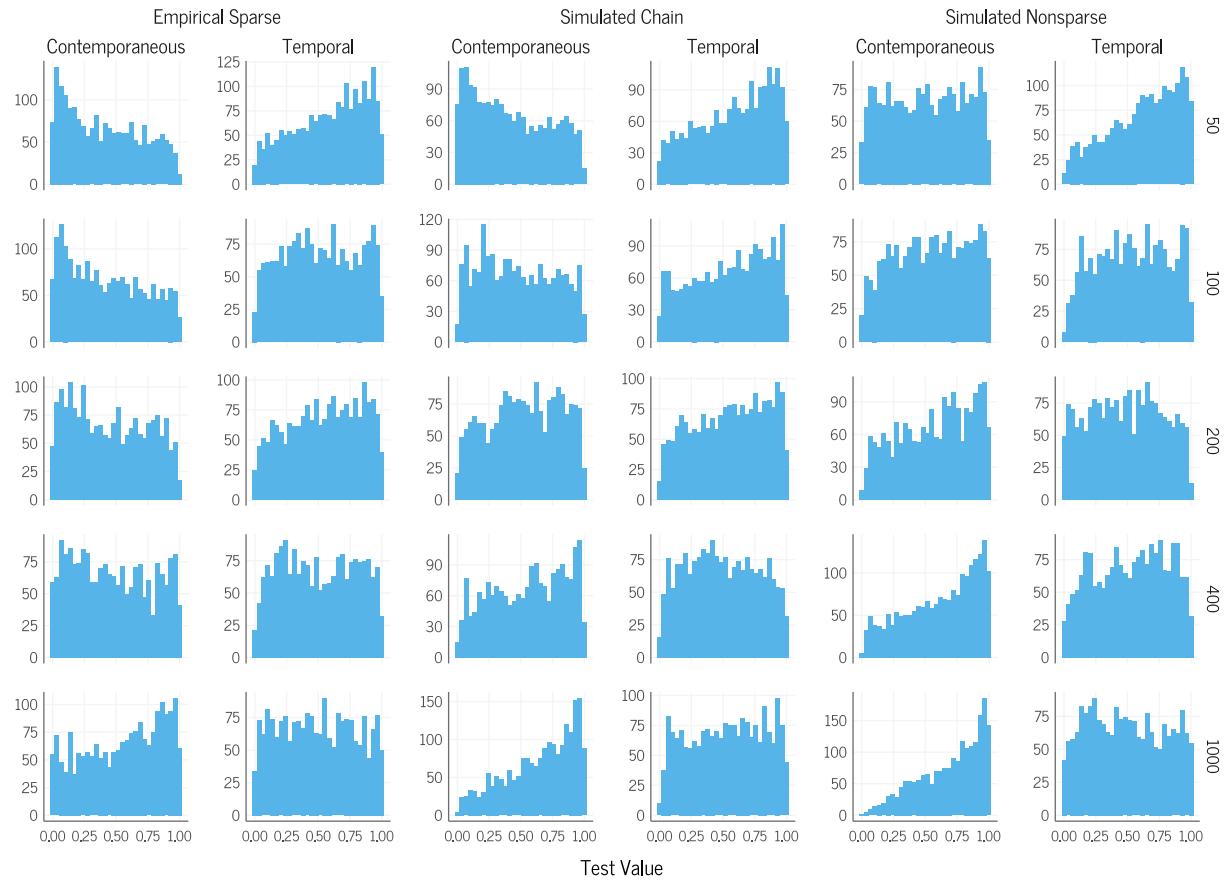
Empirical Example

In this section, we present additional analyses for the empirical example. We show multiple sampling diagnostics and the prior sensitivity of the test in the example data.

Sampling Diagnostics

Here, we report sampling diagnostics for the 40 models in the empirical example, focussing on the prior combination we used to present our results ($s_\rho = 0.25$ and $s_\beta = 0.5$). These can either be obtained by using the `convergence`-function in *BGGM* for trace and autocorrelation plots, or by using the `ess_gvar`-function in the *tsnet* to compute the effective sample size. An overview of sampling diagnostics for MCMC sampling is given in Roy (2020).

Figure 10
Distribution of Test Values under the Null for the Wide Prior.



Note. Using Frobenius norm only. X-axis displays different data-generating processes, y-axis displays different sample sizes.

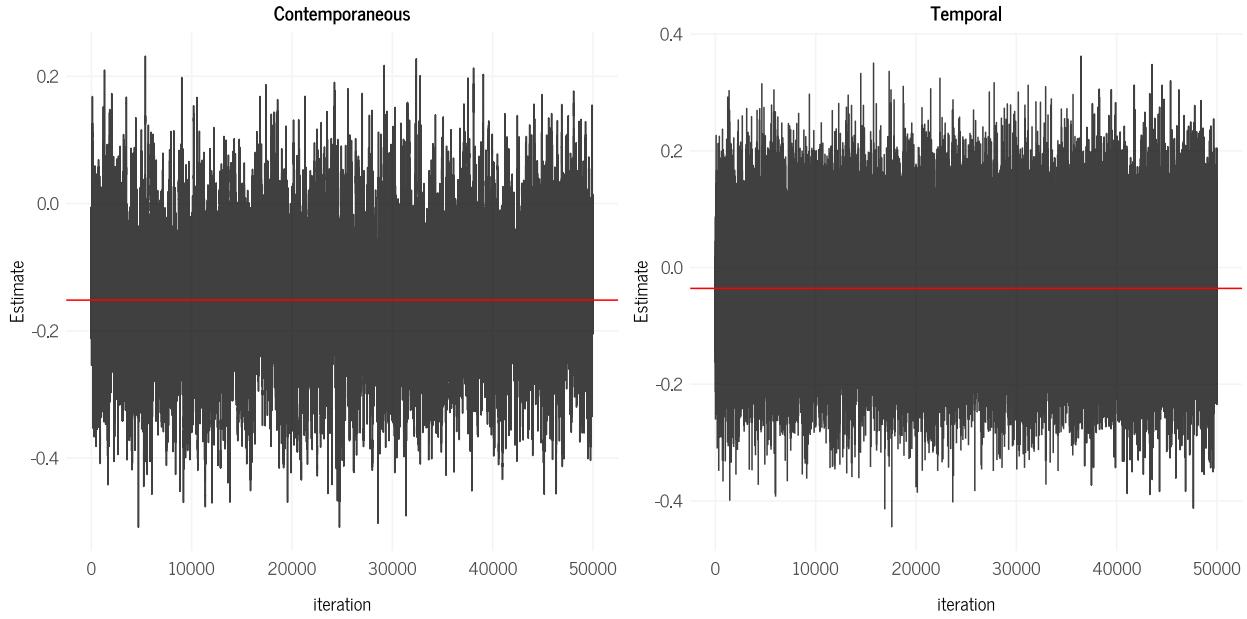
Trace Plots

Trace plots are an important MCMC diagnostic tool to check the mixing of a chain.

Trace plot shows the value of the Markov chain at each point in the sampling process.

Overall, the results here looked similar to the example in Figure 11, indicated by the resemblance of the line to a hairy caterpillar.

We include code to create all trace plots for all parameters in the electronic supplement.

**Figure 11**

Example Trace Plot for Items Content and Concentration for ID 19.

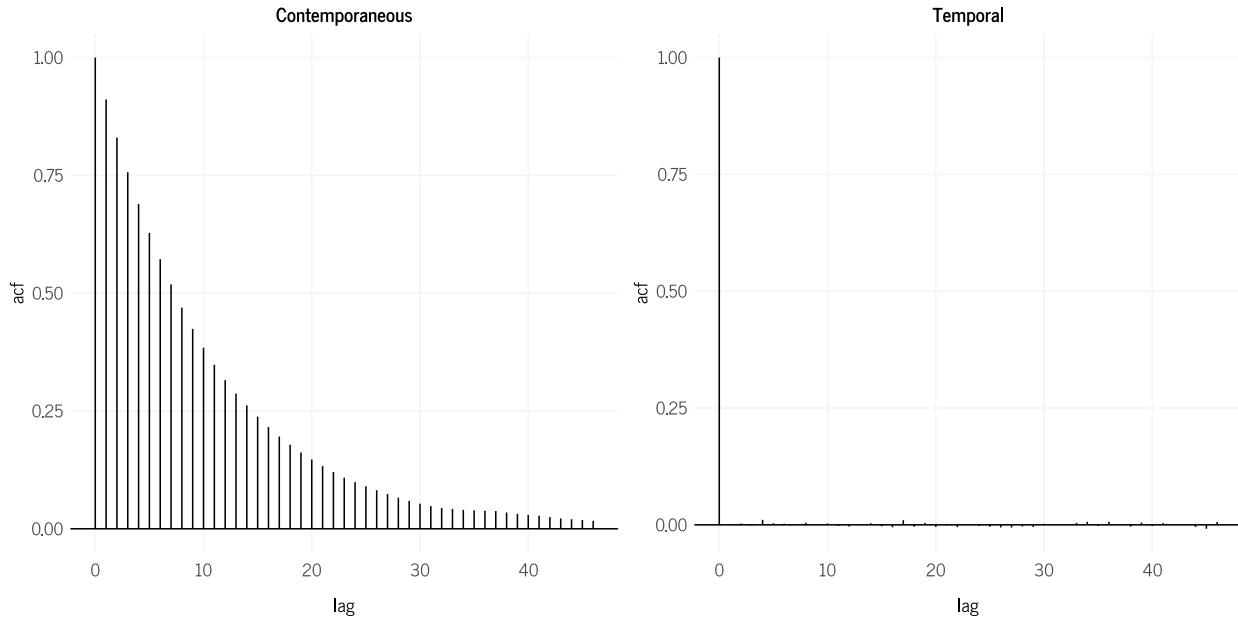
Autocorrelation

Assessing the autocorrelation of samples is important to gain insight into the efficiency or speed of the sampling as well as the quality of the resulting posterior distribution. Here, the lag- k autocorrelation is the correlation of a specific sample with a sample k values spaced apart. A high autocorrelation across larger values of k indicates slow mixing, and more iterations are needed to obtain reliable estimates. Results for the autocorrelation were very similar across individuals, but clearly different across different parameters. Coefficients of the temporal network exhibited a low autocorrelation past lag-1, which is desirable. However, coefficients of the contemporaneous network showed high autocorrelations across a range of lags. Example plots representative of this tendency are shown in Figure 12, where the values of the Autocorrelation function (ACF) on the y-axis are plotted against the lag value k on the x-axis.

We include code to create all autocorrelation plots for all parameters in the electronic supplement.

Figure 12

Example ACF Plot for Items Content and Concentration for ID 19.



Effective Sample size

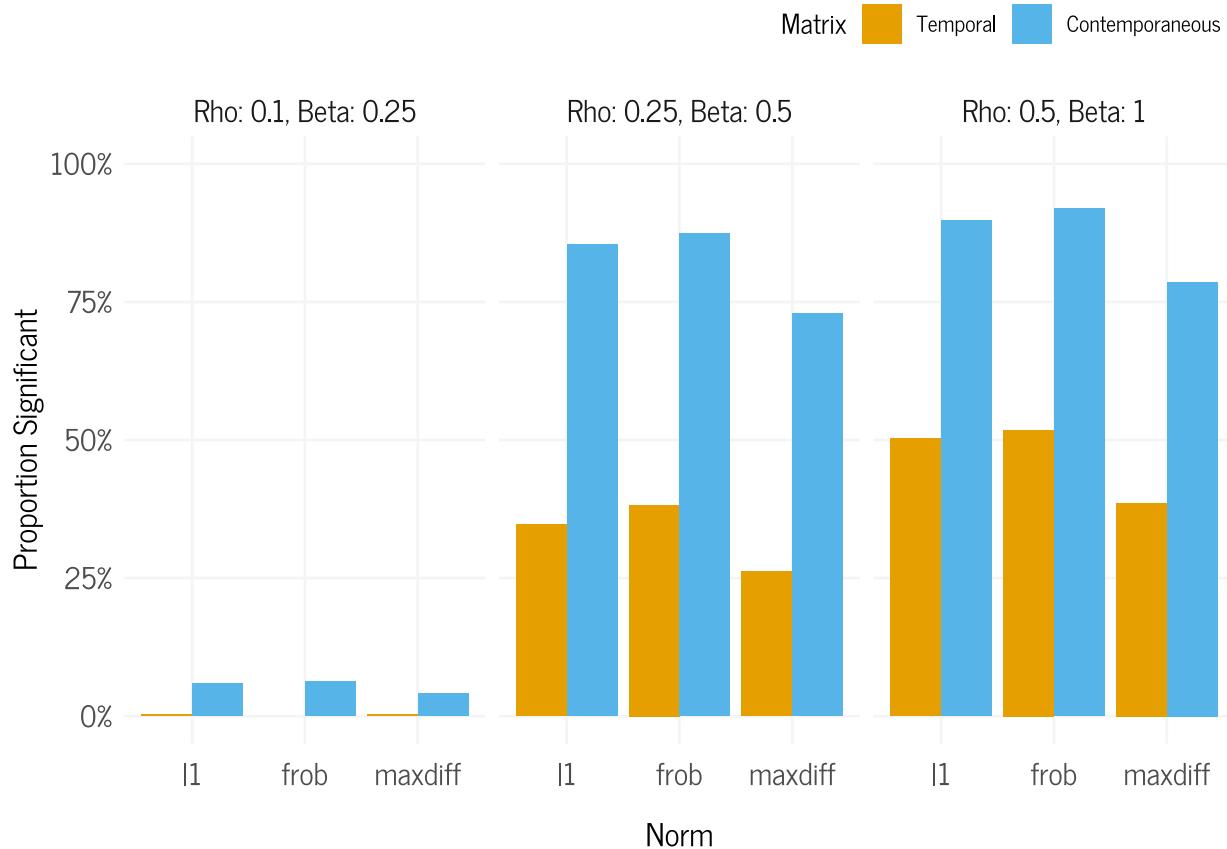
We calculated the Effective Sample Size (ESS) using the `ess_gvar`-function in the *tsnet*, which implements the calculation of the *coda*-package (Plummer et al., 2020) for *BGGM* objects. It reflects the number of uncorrelated MCMC samples equivalent to a set of correlated MCMC samples (Roy, 2020). Due to the autocorrelation in the sampling for the contemporaneous network, the effective sample size for these parameters was substantially lower than for the temporal network parameters, which prompted us to use 50.000 iterations across all simulations. Specifically, the average effective sample size across all parameters in all temporal networks was 50020.31, whereas it was 2687.32 for the contemporaneous networks. Parameters within the networks were very similar overall. Across individuals, the ESS of the temporal parameters were very similar as well, ranging from a mean individual ESS of 49868.97 to 50161.13. On the contrary, the ESS of the contemporaneous network showed more heterogeneity, ranging from a mean individual ESS of 2142.43 to 4361.70.

Prior Sensitivity

In Figure 13, we show the prior sensitivity of the number of positive test results for all 780 possible comparisons in the empirical example, computed as described in the manuscript. Clearly, using a very narrow prior prevents the detection of reliable differences between networks, as all estimates are pulled together towards zero across participants. The other two priors led to more similar results, where a wider prior leads to more positive test results. This difference is relatively more pronounced in the temporal network. We did not account for possible issues with multiple testing here.

Figure 13

Prior Sensitivity Empirical Example.



Note. X-axis shows the three different norms, different grids along the x-axis show increasingly wide priors with different prior standard deviations for the contemporaneous and temporal network. Y-axis shows the proportion of positive test values across all possible pairwise comparisons.

Session Information

This section contains the R session information for the local machine that was used to analyze the simulations as well as the Server that was used to compute the simulations.

Local Environment.

- R version 4.3.2 (2023-10-31 ucrt), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=German_Germany.utf8`, `LC_CTYPE=German_Germany.utf8`,
`LC_MONETARY=German_Germany.utf8`, `LC_NUMERIC=C`,
`LC_TIME=German_Germany.utf8`
- Time zone: `Europe/Berlin`
- TZcode source: `internal`
- Running under: `Windows 11 x64 (build 22621)`
- Matrix products: `default`
- Base packages: `base`, `datasets`, `graphics`, `grDevices`, `methods`, `stats`, `utils`
- Loaded via a namespace (and not attached): `abind` 1.4-5, `backports` 1.4.1, `bain` 0.2.10, `base64enc` 0.1-3, `bayestestR` 0.13.1, `BFpack` 1.2.3, `BGGM` 2.1.0, `boot` 1.3-28.1, `cachem` 1.0.8, `cellranger` 1.1.0, `checkmate` 2.3.1, `cli` 3.6.2, `cluster` 2.1.4, `coda` 0.19-4, `codetools` 0.2-19, `colorspace` 2.1-0, `compiler` 4.3.2, `corpcor` 1.6.10, `cowplot` 1.1.3, `curl` 5.2.0, `data.table` 1.14.10, `datawizard` 0.9.1, `DEoptimR` 1.1-3, `digest` 0.6.34, `distributional` 0.3.2, `doRNG` 1.8.6, `dplyr` 1.1.4, `effectsize` 0.8.6, `emmeans` 1.10.0, `ergm` 4.6.0, `estimability` 1.4.1, `evaluate` 0.23, `extraDistr` 1.10.0, `fansi` 1.0.6, `farver` 2.1.1, `fastmap` 1.1.1, `fdrtool` 1.2.17, `forcats` 1.0.0, `foreach` 1.5.2, `foreign` 0.8-85, `Formula` 1.2-5, `generics` 0.1.3, `GGally` 2.2.0, `ggdist` 3.3.1, `ggh4x` 0.2.8, `ggokabeito` 0.1.0, `ggplot2` 3.4.4, `ggridges` 0.5.6, `ggstats` 0.5.1, `glasso` 1.11, `glmnet` 4.1-8, `glue` 1.7.0, `graphicalVAR` 0.3.3, `grid` 4.3.2, `gridExtra` 2.3, `gsl` 2.1-8, `gttable` 0.3.4, `gtools` 3.9.5, `here` 1.0.1, `Hmisc` 5.1-1,

htmlTable 2.4.2, htmltools 0.5.7, htmlwidgets 1.6.4, igraph 1.6.0, inline 0.3.19, insight 0.19.7, iterators 1.0.14, janitor 2.2.0, jpeg 0.1-10, jsonlite 1.8.8, knitr 1.45, lattice 0.21-9, lavaan 0.6-17, lifecycle 1.0.4, lme4 1.1-35.1, loo 2.6.0, lpSolveAPI 5.5.2.0-17.11, lubridate 1.9.3, magick 2.8.2, magrittr 2.0.3, MASS 7.3-60, Matrix 1.6-5, matrixcalc 1.0-6, matrixStats 1.2.0, memoise 2.0.1, mgcv 1.9-0, minqa 1.2.6, mnormt 2.1.1, munsell 0.5.0, mvtnorm 1.2-4, network 1.18.2, nlme 3.1-163, nloptr 2.0.3, nnet 7.3-19, numDeriv 2016.8-1.1, parallel 4.3.2, parameters 0.21.3, patchwork 1.2.0, pbapply 1.7-2, pbivnorm 0.6.0, pillar 1.9.0, pkgbuild 1.4.3, pkgconfig 2.0.3, plyr 1.8.9, png 0.1-8, pracma 2.4.4, psych 2.4.1, purrr 1.0.2, qgraph 1.9.8, QRM 0.4-31, quadprog 1.5-8, QuickJSR 1.1.0, R6 2.5.1, rbibutils 2.2.16, RColorBrewer 1.1-3, Rcpp 1.0.12, RcppParallel 5.1.7, Rdpack 2.6, readxl 1.4.3, reshape 0.8.9, reshape2 1.4.4, rlang 1.1.3, rle 0.9.2, rmarkdown 2.25, rngtools 1.5.2, robustbase 0.99-1, rpart 4.1.21, rprojroot 2.0.4, rstan 2.32.5, rstudioapi 0.15.0, sandwich 3.1-0, scales 1.3.0, shape 1.4.6, showtext 0.9-6, showtextdb 3.0, sna 2.7-2, snakecase 0.11.1, splines 4.3.2, StanHeaders 2.32.5, statnet.common 4.9.0, stats4 4.3.2, stringi 1.8.3, stringr 1.5.1, survival 3.5-7, sysfonts 0.8.8, tibble 3.2.1, tidyverse 1.3.1, tidyselect 1.2.0, timechange 0.3.0, timeDate 4032.109, timeSeries 4032.109, tools 4.3.2, trust 0.1-8, tsnet 0.1.0, utf8 1.2.4, V8 4.4.1, vctrs 0.6.5, xfun 0.41, xtable 1.8-4, yaml 2.3.8, zoo 1.8-12

Server Environment.

- R version 4.3.0 (2023-04-21), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=de_DE.UTF-8, LC_NUMERIC=C, LC_TIME=de_DE.UTF-8, LC_COLLATE=de_DE.UTF-8, LC_MONETARY=de_DE.UTF-8, LC_MESSAGES=de_DE.UTF-8, LC_PAPER=de_DE.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=de_DE.UTF-8, LC_IDENTIFICATION=C
- Time zone: Etc/UTC

- TZcode source: `system (glibc)`
- Running under: Ubuntu 20.04.6 LTS
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnublas/libblas.so.3.9.0`
- LAPACK: `/usr/lib/x86_64-linux-gnulapack/liblapack.so.3.9.0`
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, utils
- Other packages: BGGM 2.0.4, cowplot 1.1.1, doParallel 1.0.17, doRNG 1.8.6, dplyr 1.1.2,forcats 1.0.0, foreach 1.5.2, ggh4x 0.2.4, ggplot2 3.4.2, graphicalVAR 0.3.1, here 1.0.1, iterators 1.0.14, lubridate 1.9.2, Matrix 1.5-4.1, mgm 1.2-13, mlVAR 0.5.1, mvtnorm 1.2-1, purrr 1.0.1, readr 2.1.4, reshape2 1.4.4, rngtools 1.5.2, rrapply 1.2.6, stringr 1.5.0, sysfonts 0.8.8, tibble 3.2.1, tidyverse 2.0.0
- Loaded via a namespace (and not attached): abind 1.4-5, arm 1.13-1, backports 1.4.1, bain 0.2.8, base64enc 0.1-3, BFpack 1.0.0, boot 1.3-28.1, checkmate 2.2.0, cli 3.6.1, cluster 2.1.4, clusterGeneration 1.3.7, coda 0.19-4, codetools 0.2-19, colorspace 2.1-0, compiler 4.3.0, corpcor 1.6.10, data.table 1.14.8, digest 0.6.31, evaluate 0.21, extraDistr 1.9.1, fansi 1.0.4, fastDummies 1.6.3, fastmap 1.1.1, fdrtool 1.2.17, foreign 0.8-84, Formula 1.2-5, generics 0.1.3, GGally 2.1.2, ggokabeito 0.1.0, ggridges 0.5.4, glasso 1.11, glmnet 4.1-7, glue 1.6.2, grid 4.3.0, gridExtra 2.3, gsubfn 0.7, gtable 0.3.3, gtools 3.9.4, Hmisc 5.1-0, hms 1.1.3, htmlTable 2.4.1, htmltools 0.5.5, htmlwidgets 1.6.2, httr 1.4.6, igraph 1.4.3, jpeg 0.1-10, knitr 1.43, lattice 0.21-8, lavaan 0.6-15, lifecycle 1.0.3, lme4 1.1-33, magrittr 2.0.3, MASS 7.3-60, minqa 1.2.5, mnormt 2.1.1, MplusAutomation 1.1.0, munsell 0.5.0, network 1.18.1, nlme 3.1-162, nloptr 2.0.3, nnet 7.3-19, pander 0.6.5, patchwork 1.1.2, pbapply 1.7-0, pbivnorm 0.6.0, pillar 1.9.0, pkgconfig 2.0.3, plyr 1.8.8, png 0.1-8, pracma 2.4.2,

proto 1.0.0, psych 2.3.3, qgraph 1.9.5, quadprog 1.5-8, R6 2.5.1, rbibutils 2.2.13,
RColorBrewer 1.1-3, Rcpp 1.0.10, Rdpack 2.4, reshape 0.8.9, rlang 1.1.1,
rmarkdown 2.22, rpart 4.1.19, rprojroot 2.0.3, rstudioapi 0.14, scales 1.2.1, shape 1.4.6,
sna 2.7-1, splines 4.3.0, statnet.common 4.9.0, stats4 4.3.0, stringi 1.7.12, survival 3.5-5,
texreg 1.38.6, tidyselect 1.2.0, timechange 0.2.0, tools 4.3.0, tzdb 0.4.0, utf8 1.2.3,
vctrs 0.6.2, withr 2.5.0, xfun 0.39, xtable 1.8-4, yaml 2.3.7

References

- Barnard, J., McCulloch, R., & Meng, X.-L. (2000). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, 10(4), 1281–1311. Retrieved December 15, 2023, from <https://www.jstor.org/stable/24306780>
- Chen, J., & Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3), 759–771. <https://doi.org/10.1093/biomet/asn034>
- Epskamp, S., Kruis, J., & Marsman, M. (2017). Estimating psychopathological networks: Be careful what you wish for. *PLOS ONE*, 12(6), e0179891. <https://doi.org/10.1371/journal.pone.0179891>
- Epskamp, S., van Borkulo, C. D., van der Veen, D. C., Servaas, M. N., Isvoranu, A.-M., Riese, H., & Cramer, A. O. J. (2018). Personalized network modeling in psychopathology: The importance of contemporaneous and temporal connections. *Clinical Psychological Science*, 6(3), 416–427. <https://doi.org/10.1177/2167702617744325>
- Epskamp, S., Waldorp, L. J., Möttus, R., & Borsboom, D. (2018). The Gaussian graphical model in cross-sectional and time-series data. *Multivariate behavioral research*, 53(4), 453–480.
- Fried, E. I., Papanikolaou, F., & Epskamp, S. (2021). Mental health and social contact during the covid-19 pandemic: An ecological momentary assessment study. *Clinical Psychological Science*, 10(2), 340–354. <https://doi.org/10.1177/21677026211017839>
- Hoekstra, R. H. A., Epskamp, S., & Borsboom, D. (2022). Heterogeneity in individual network analysis: Reality or illusion? *Multivariate Behavioral Research*, 58(4), 762–786. <https://doi.org/10.1080/00273171.2022.2128020>

- Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9), 1989–2001. <https://doi.org/10.1016/j.jmva.2009.04.008>
- Mansueto, A. C., Wiers, R. W., van Weert, J. C. M., Schouten, B. C., & Epskamp, S. (2023). Investigating the feasibility of idiographic network models. *Psychological Methods*, 28(5), 1052–1068. <https://doi.org/10.1037/met0000466.supp>
- Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, 38(11), 2074–2102. <https://doi.org/10.1002/sim.8086>
- Mulder, J., & Pericchi, L. R. (2018). The Matrix-F prior for estimating and testing covariance matrices. *Bayesian Analysis*, 13(4), 1193–1214. <https://doi.org/10.1214/17-BA1092>
- Plummer, M., Best, N., Cowles, K., Vines, K., Sarkar, D., Bates, D., Almond, R., & Magnusson, A. (2020, September 30). *Coda: Output Analysis and Diagnostics for MCMC* (Version 0.19-4). Retrieved April 19, 2023, from <https://CRAN.R-project.org/package=coda>
- Rothman, A. J., Levina, E., & Zhu, J. (2010). Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4), 947–962. Retrieved March 17, 2023, from <https://www.jstor.org/stable/25765382>
- Roy, V. (2020). Convergence diagnostics for Markov Chain Monte Carlo. *Annual Review of Statistics and Its Application*, 7(1), 387–412. <https://doi.org/10.1146/annurev-statistics-031219-041300>
- Schuurman, N. K., Grasman, R. P. P. P., & Hamaker, E. L. (2016). A comparison of inverse-wishart prior specifications for covariance matrices in multilevel autoregressive models. *Multivariate Behavioral Research*, 51(2-3), 185–206. <https://doi.org/10.1080/00273171.2015.1065398>

- Sekulovski, N., Keetelaar, S., Huth, K., Wagenmakers, E.-J., Bork, R. van, Bergh, D. van den, & Marsman, M. (2023, April 19). *Testing conditional independence in psychometric networks: An analysis of three bayesian methods.* <https://doi.org/10.31234/osf.io/ch7a2>
- Siepe, B. S., Bartoš, F., Morris, T., Boulesteix, A.-L., Heck, D. W., & Pawel, S. (2023, October 31). *Simulation Studies for Methodological Research in Psychology: A Standardized Template for Planning, Preregistration, and Reporting.* <https://doi.org/10.31234/osf.io/ufgy6>
- Stan Development Team. (2024, October 1). *Rstan: R Interface to Stan* (Version 2.32.5). Retrieved January 24, 2024, from <https://mc-stan.org/cmdstanr/>
- Williams, D. R., & Mulder, J. (2020). Bayesian hypothesis testing for Gaussian graphical models: Conditional independence and order constraints. *Journal of Mathematical Psychology*, 99, 102441. <https://doi.org/10.1016/j.jmp.2020.102441>
- Williams, D. R., Rast, P., Pericchi, L. R., & Mulder, J. (2020). Comparing gaussian graphical models with the posterior predictive distribution and Bayesian model selection. *Psychological Methods*, 25(5), 653–672. <https://doi.org/10.1037/met0000254>