Change Point Detection on A Separable Model for Dynamic Networks

Anonymous authors
Paper under double-blind review

Abstract

This paper studies the unsupervised change point detection problem in time series of networks using the Separable Temporal Exponential-family Random Graph Model (STERGM). Inherently, dynamic network patterns can be complex due to dyadic and temporal dependence, and change points detection can identify the discrepancies in the underlying data generating processes to facilitate downstream analysis. Moreover, the STERGM that utilizes network statistics to represent the structural patterns is a flexible and parsimonious model to fit dynamic networks. We propose a new estimator derived from the Alternating Direction Method of Multipliers (ADMM) procedure and Group Fused Lasso (GFL) regularization to simultaneously detect multiple time points, where the parameters of a time-heterogeneous STERGM have changed. We also provide a Bayesian information criterion for model selection and an R package `CPDstergm` to implement the proposed method. Experiments on simulated and real data show good performance of the proposed framework.

1 Introduction

Networks are often used to describe relational phenomena that are not limited merely to the attributes of individuals as tabular data. In an investigation of the transmission of COVID-19, Fritz et al. (2021) used networks to represent human mobility and forecast disease incidents. The study of physical connections, beyond the health status of individuals, permits policymakers to implement preventive measures effectively and to allocate healthcare resources efficiently. Yet relations progress in time, and dynamic relational phenomena are occasionally aggregated into a static network for analysis. To this end, temporal models for dynamic networks are in high demand to study the evolution of relational phenomena.

In recent decades, a plethora of temporal models has been proposed for dynamic networks analysis. Snijders (2001), Snijders (2005), and Snijders et al. (2010) developed a Stochastic Actor-Oriented Model, which is driven by the actor's perspective to make or withdraw ties to other actors in a network. Hoff et al. (2002), Sarkar & Moore (2005), Sewell & Chen (2015), and Sewell & Chen (2016) presented latent space models, by assuming the edges between actors are more likely when they are closer in the latent Euclidean space. Matias & Miele (2017), Ludkin et al. (2018), and Pensky (2019) investigated the dynamic Stochastic Block Model (SBM), and Jiang et al. (2020) developed an autoregressive SBM to characterize the evolution of communities. Kolar et al. (2010) focused on recovering the latent time-varying graph structures of Markov Random Fields from serial observations of nodal attributes. Furthermore, the Exponential-family Random Graph Model (ERGM) that uses local forces to shape global structures (Hunter et al., 2008b) is a promising model for networks with dependent ties. Hanneke et al. (2010) defined a Temporal ERGM (TERGM), by conditioning on previous networks in the network statistics of an ERGM. Desmarais & Cranmer (2012b) proposed a bootstrap approach to maximize the pseudo-likelihood of the TERGM and assess uncertainty. In general, network evolution concerns the rate at which edges form and dissolve. Demonstrated in Krivitsky & Handcock (2014), these two factors can be mutually interfering, making the dynamic models used in the literature difficult to interpret. Posing that the underlying reasons that result in dyad formation are different from those that result in dyad dissolution, Krivitsky & Handcock (2014) proposed a Separable Temporal ERGM (STERGM) to dissect the entanglement with two conditionally independent models.

In time series analysis, change point detection plays a central role in identifying the discrepancies in the underlying data generating processes over time. In reality, network evolution is usually time-heterogeneous. Without taking the structural changes across dynamic networks into consideration, learning from the time series may not be meaningful, by confounding the network effects before and after a change occurs. As relational phenomena are studied in numerous domains, it is practical for researchers to first localize the change points, and then analyze the network effects within segments, rather than overlooking the time points where the network patterns have substantially changed.

There has also been an increasing interest in studying the unsupervised change point detection problem for dynamic networks. Wang et al. (2013) focused on the Stochastic Block Model time series, and Wang et al. (2021) studied a sequence of inhomogeneous Bernoulli networks. Larroca et al. (2021), Marenco et al. (2022), and Madrid Padilla et al. (2022) considered a sequence of Random Dot Product Graphs that are both dyadic and temporal dependent. Methodologically, Chen & Zhang (2015) and Chu & Chen (2019) developed a graph-based approach to delineate the distributional differences before and after a change point, and Chen (2019) utilized the nearest neighbor information to detect the changes in an online framework. Zhao et al. (2019) proposed a two-step approach that consists of an initial graphon estimation followed by a screening algorithm, Song & Chen (2022b) exploited the features in high dimensions via a kernel-based method, and Chen et al. (2020a) employed embedding methods to detect both anomalous graphs and anomalous vertices. Chen et al. (2024) and Athreya et al. (2024) developed a model for network time series based on a latent position process, using spectral estimates of the Euclidean mirror to detect first-order change points. Zhang et al. (2024) combined Variational Graph Auto-Encoder and Gaussian Mixture Model, and Kei et al. (2024) focused on graph representation learning for change point detection in dynamic networks. Moreover, Liu et al. (2018) introduced an eigenvector-based method to reveal the change and persistence in the gene communities for a developing brain. Bybee & Atchadé (2018) focused on a Gaussian Graphical Model to detect the change points in the covariance structure of the Standard and Poor's 500. Ondrus et al. (2021) proposed a factorized binary search method to understand brain connectivity from the functional Magnetic Resonance Imaging time series data.

Allowing for interpretable network statistics to determine the structural changes for the detection, we make the following contributions in the proposed framework:

- To detect multiple change points from a sequence of networks, we fit a time-heterogeneous STERGM while penalizing the sum of Euclidean norms of the sequential parameter differences. Essentially, we impose a Group Fused Lasso (GFL) regularization on the model parameters, smoothing out minor variation and highlighting significant structural changes. We also derive an Alternating Direction Method of Multipliers (ADMM) procedure to solve the resulting optimization problem.

- We exploit the practicality of STERGM, which manages dyad formation and dissolution separately, to capture the structural changes in network evolution realistically. The flexibility of STERGM and the extensive selection of network statistics also boost the power of the proposed method. Moreover, we provide a Bayesian information criterion for model selection, and we develop an R package `CPDstergm` to implement the proposed method.

- We simulate dynamic networks to imitate realistic social interactions, and our method can achieve greater accuracy on the networks that are both dyadic and temporal dependent. Furthermore, we punctually detect the winter and spring vacations with the MIT cellphone data (Eagle & Pentland, 2006) and we detect three significant financial events during the 2008 worldwide economic crisis from the stock market data analyzed in James & Matteson (2015).

The rest of the paper is organized as follows. In Section 2, we review the STERGM for dynamic networks. In Section 3, we present the likelihood-based objective function with Group Fused Lass regularization, and we derive an Alternating Direction Method of Multipliers to solve the optimization problem. In Section 4, we discuss change points localization after parameter learning, along with model selection. In Section 5, we implement our method on simulated and real data. In Section 6, we conclude our work with a discussion on the limitation and potential future developments.

2 STERGM Change Point Model

2.1 ERGM

For a node set $N = \{1, 2, \cdots, n\}$, we can use a network $\boldsymbol{y} \in \mathcal{Y} = \{0,1\}^{n\times n}$ to represent the potential relations for all pairs $(i,j) \in \mathbb{Y} = N \times N$. The network $\boldsymbol{y}$ has dyad $\boldsymbol{y}_{ij} \in \{0,1\}$ to indicate the absence or presence of a relation between node i and node j. The relations in a network can be either directed or undirected, where an undirected network has $\boldsymbol{y}_{ij} = \boldsymbol{y}_{ji}$ for all $(i,j) \in \mathbb{Y}$.

The probabilistic formulation of an Exponential-family Random Graph Model (ERGM) is

$$\boldsymbol{y} \sim P(\boldsymbol{y}; \boldsymbol{\theta}) = \exp[\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{y}) - \psi(\boldsymbol{\theta})] \tag{1}$$

where $\boldsymbol{g}(\boldsymbol{y})$, with $\boldsymbol{g} : \mathcal{Y} \to \mathbb{R}^p$, is a vector of network statistics; $\boldsymbol{\theta} \in \mathbb{R}^p$ is a vector of parameters; $\exp[\psi(\boldsymbol{\theta})] = \sum_{\boldsymbol{y}\in\mathcal{Y}} \exp[\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{y})]$ is the normalizing constant. The network statistics $\boldsymbol{g}(\boldsymbol{y})$ may depend on nodal attributes $\boldsymbol{x}$. For notational simplicity, we omit the dependence of $\boldsymbol{g}(\boldsymbol{y})$ on $\boldsymbol{x}$.

With a surrogate as in Besag (1974); Strauss & Ikeda (1990); Van Duijn et al. (2009); Desmarais & Cranmer (2012b); Hummel et al. (2012), the log-likelihood of an ERGM in (1) can be approximated as

$$l(\boldsymbol{\theta})_{\text{ERGM}} = \sum_{(i,j)\in\mathbb{Y}} \Big\{ \boldsymbol{y}_{ij}[\boldsymbol{\theta} \cdot \Delta\boldsymbol{g}_{ij}(\boldsymbol{y})] - \log\big\{1 + \exp[\boldsymbol{\theta} \cdot \Delta\boldsymbol{g}_{ij}(\boldsymbol{y})]\big\} \Big\}$$

where the change statistics $\Delta\boldsymbol{g}_{ij}(\boldsymbol{y}) \in \mathbb{R}^p$ denote the change in $\boldsymbol{g}(\boldsymbol{y})$ when $\boldsymbol{y}_{ij}$ changes from 0 to 1, while rest of the network remains the same. This formulation is called the logarithm of the pseudo-likelihood, and it is helpful in ERGM parameter estimation. Moreover, the Hessian matrix of $-l(\boldsymbol{\theta})_{\text{ERGM}}$ is

$$H(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathbb{Y}} h\big(\boldsymbol{\theta} \cdot \Delta\boldsymbol{g}_{ij}(\boldsymbol{y})\big) \cdot \big[1 - h\big(\boldsymbol{\theta} \cdot \Delta\boldsymbol{g}_{ij}(\boldsymbol{y})\big)\big] \cdot \big[\Delta\boldsymbol{g}_{ij}(\boldsymbol{y}) \Delta\boldsymbol{g}_{ij}(\boldsymbol{y})^\top\big]$$

where $h(x) = 1/(1+\exp(-x))$ is the sigmoid function. Since $h(\boldsymbol{\theta} \cdot \Delta\boldsymbol{g}_{ij}(\boldsymbol{y})) \in (0,1)$ and $\Delta\boldsymbol{g}_{ij}(\boldsymbol{y}) \Delta\boldsymbol{g}_{ij}(\boldsymbol{y})^\top \in \mathbb{R}^{p\times p}$ is positive semi-definite, the Hessian matrix $H(\boldsymbol{\theta}) = \nabla^2_{\boldsymbol{\theta}} - l(\boldsymbol{\theta})_{\text{ERGM}}$ is positive semi-definite. In other words, the negative logarithm of the pseudo-likelihood is convex with respect to $\boldsymbol{\theta} \in \mathbb{R}^p$. Next, we introduce the Separable Temporal ERGM (STERGM) used in our change point model.

2.2 STERGM

For network time series, network evolution concerns (1) incidence: how often new ties are formed, and (2) duration: how long old ties last since they were formed. Donnat & Holmes (2018), Goyal & De Gruttola (2020), and Jiang et al. (2020) pointed out that modeling snapshots of networks may give limited information about the transitions between consecutive networks. To address this issue, Krivitsky & Handcock (2014) designed two intermediate networks, formation network and dissolution network, to reflect the incidence and duration. In particular, the incidence can be measured by dyad formation, and the duration can be traced by dyad dissolution. Many applications on real-world data support the separable assumption for dynamic networks (Broekel & Bednarz, 2018; Uppala & Handcock, 2020; Ando et al., 2025).

Let $\boldsymbol{y}^t \in \mathcal{Y}^t = \{0,1\}^{n\times n}$ be a network observed at a discrete time point t. The formation network $\boldsymbol{y}^{+,t} \in \mathcal{Y}^{+,t}$ is obtained by attaching the edges that formed at time t to $\boldsymbol{y}^{t-1}$, and $\mathcal{Y}^{+,t} = \{\boldsymbol{y} \in \mathcal{Y}^t : \boldsymbol{y} \supseteq \boldsymbol{y}^{t-1}\}$. The dissolution network $\boldsymbol{y}^{-,t} \in \mathcal{Y}^{-,t}$ is obtained by deleting the edges that dissolved at time t from $\boldsymbol{y}^{t-1}$, and $\mathcal{Y}^{-,t} = \{\boldsymbol{y} \in \mathcal{Y}^t : \boldsymbol{y} \subseteq \boldsymbol{y}^{t-1}\}$. We also use the notation from Kei et al. (2023) to specify the respective formation and dissolution networks between time $t-1$ and time t as

$$\boldsymbol{y}^{+,t}_{ij} = \max(\boldsymbol{y}^{t-1}_{ij}, \boldsymbol{y}^t_{ij}) \;\text{ and }\; \boldsymbol{y}^{-,t}_{ij} = \min(\boldsymbol{y}^{t-1}_{ij}, \boldsymbol{y}^t_{ij})$$

for all $(i,j) \in \mathbb{Y}$. In summary, $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ incorporate the dependence on $\boldsymbol{y}^{t-1}$ through construction, and they can be considered as two latent networks recovered from both $\boldsymbol{y}^{t-1}$ and $\boldsymbol{y}^t$ to emphasize the network transition from time $t-1$ to time t.

Posing that the underlying factors that result in edge formation are different from those that result in edge dissolution, Krivitsky & Handcock (2014) proposed the STERGM to dissect the evolution between consecutive networks. Assuming $\boldsymbol{y}^{+,t}$ is conditionally independent of $\boldsymbol{y}^{-,t}$ given $\boldsymbol{y}^{t-1}$, the STERGM for $\boldsymbol{y}^t$ conditional on $\boldsymbol{y}^{t-1}$ is

$$P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^t) = P(\boldsymbol{y}^{+,t}|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^{+,t}) \times P(\boldsymbol{y}^{-,t}|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^{-,t}) \tag{2}$$

with the respective formation and dissolution models:

$$\begin{aligned} P(\boldsymbol{y}^{+,t}|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^{+,t}) &= \exp[\boldsymbol{\theta}^{+,t} \cdot \boldsymbol{g}^{+}(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\theta}^{+,t}, \boldsymbol{y}^{t-1})], \\ P(\boldsymbol{y}^{-,t}|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^{-,t}) &= \exp[\boldsymbol{\theta}^{-,t} \cdot \boldsymbol{g}^{-}(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\theta}^{-,t}, \boldsymbol{y}^{t-1})]. \end{aligned}$$

The parameter $\boldsymbol{\theta}^t = (\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t}) \in \mathbb{R}^p$ is a concatenation of $\boldsymbol{\theta}^{+,t} \in \mathbb{R}^{p_1}$ and $\boldsymbol{\theta}^{-,t} \in \mathbb{R}^{p_2}$ such that $p_1 + p_2 = p$.

Notably, the normalizing constant in the formation model at a time point t:

$$\exp[\psi(\boldsymbol{\theta}^{+,t}, \boldsymbol{y}^{t-1})] = \sum_{\boldsymbol{y}^{+,t} \in \mathcal{Y}^{+,t}} \exp[\boldsymbol{\theta}^{+,t} \cdot \boldsymbol{g}^{+}(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})]$$

is a sum over all possible networks in $\mathcal{Y}^{+,t}$, and that in the dissolution model is similar except for notational difference. Measuring these normalizing constants can be computationally intractable when the number of nodes n is large (Hunter & Handcock, 2006). Thus, for the change point detection problem described in Section 3, we adopt the pseudo-likelihood of an ERGM to estimate the parameters. For a network modeling problem, other parameter estimation methods exploit MCMC sampling (Geyer & Thompson, 1992; Krivitsky, 2017) or Bayesian inference (Caimo & Friel, 2011; Thiemichen et al., 2016) to circumvent the intractability of the normalizing constants.

Specifically, the logarithm of the pseudo-likelihood of a time-heterogeneous STERGM $\prod_{t=2}^{T} P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^t)$ is formulated as

$$\begin{aligned} l(\boldsymbol{\theta}) = &\sum_{t=2}^{T} \sum_{(i,j)\in\mathbb{Y}} \Big\{ \boldsymbol{y}_{ij}^{+,t} [\boldsymbol{\theta}^{+,t} \cdot \Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})] - \log\big\{1 + \exp[\boldsymbol{\theta}^{+,t} \cdot \Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})]\big\} \Big\} + \\ &\sum_{t=2}^{T} \sum_{(i,j)\in\mathbb{Y}} \Big\{ \boldsymbol{y}_{ij}^{-,t} [\boldsymbol{\theta}^{-,t} \cdot \Delta\boldsymbol{g}_{ij}^{-}(\boldsymbol{y}^{-,t})] - \log\big\{1 + \exp[\boldsymbol{\theta}^{-,t} \cdot \Delta\boldsymbol{g}_{ij}^{-}(\boldsymbol{y}^{-,t})]\big\} \Big\} \end{aligned} \tag{3}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^T) \in \mathbb{R}^{\tau \times p}$ with $\tau = T - 1$. The change statistics $\Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})$ denote the change in $\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t})$ when $\boldsymbol{y}_{ij}^{+,t}$ changes from 0 to 1, while rest of the $\boldsymbol{y}^{+,t}$ remains the same. The $\Delta\boldsymbol{g}_{ij}^{-}(\boldsymbol{y}^{-,t})$ is defined similarly except for notational difference. Since $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ inherit the dependence on $\boldsymbol{y}^{t-1}$ by construction, we use the implicit dynamic terms, $\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t})$ with $\boldsymbol{g}^{+} : \mathcal{Y}^{+,t} \to \mathbb{R}^{p_1}$ and $\boldsymbol{g}^{-}(\boldsymbol{y}^{-,t})$ with $\boldsymbol{g}^{-} : \mathcal{Y}^{-,t} \to \mathbb{R}^{p_2}$, as discussed in Krivitsky & Handcock (2014).

The $l(\boldsymbol{\theta})$ in (3) is an approximation to the log-likelihood of (2) for $t = 2, \ldots, T$. We use the pseudo-likelihood for the optimization problem defined in Section 3 because it is computationally feasible comparing to using MCMC sampling or Bayesian inference. Throughout, the number of rows in $\boldsymbol{\theta}$ is $\tau = T - 1$ instead of T due to the transition probability $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^t)$ that is conditional on the previous network. Although $\boldsymbol{y}^t$ can be conditioned on more previous networks, we only discuss STERGM under the first-order Markov assumption in this article.

We now define the change points to be detected in terms of the parameters in STERGM. Let $\{B_k\}_{k=0}^{K+1} \subset \{1, 2, \ldots, T\}$ be a collection of ordered change points with $1 = B_0 < B_1 < \cdots < B_K < B_{K+1} = T$ such that

$$\begin{aligned} \boldsymbol{\theta}^{B_k} = \boldsymbol{\theta}^{B_k+1} = \cdots = \boldsymbol{\theta}^{B_{k+1}-1}, \quad k = 0, \ldots, K, \\ \boldsymbol{\theta}^{B_k} \neq \boldsymbol{\theta}^{B_{k+1}}, \quad k = 0, \ldots, K-1, \quad \text{and} \quad \boldsymbol{\theta}^{B_{K+1}} = \boldsymbol{\theta}^{B_K}. \end{aligned}$$

Our goal is to recover the collection $\{B_k\}_{k=1}^{K}$ from a sequence of observed networks $\{\boldsymbol{y}^t\}_{t=1}^{T}$ where the number of change points K is also unknown. Intuitively, the consecutive parameters $\boldsymbol{\theta}^t$ and $\boldsymbol{\theta}^{t+1}$ are similar when no change occurs, but one or more components in $\boldsymbol{\theta}^{B_{k+1}} \in \mathbb{R}^p$ can be different from $\boldsymbol{\theta}^{B_k} \in \mathbb{R}^p$ after a change happens. For this setting, we present our method in the next section.

3 Group Fused Lasso for STERGM

3.1 Optimization Problem

Inspired by Vert & Bleakley (2010) and Bleakley & Vert (2011), we propose the following estimator for our change point detection problem:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\arg\min}\ -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{\theta}_{i+1,.} - \boldsymbol{\theta}_{i,.}\|_2}{\boldsymbol{d}_i} \tag{4}$$

where $l(\boldsymbol{\theta})$ is formulated by (3). The Group Fused Lasso penalty expressed as the sum of Euclidean norms encourages sparsity of the parameter differences, while enforcing multiple components in $\boldsymbol{\theta}_{i+1,j} - \boldsymbol{\theta}_{i,j}$ across $j = 1, \ldots, p$ to change at the same group i. This is an effect that cannot be achieved with the ℓ_1 penalty of the differences. Along with the user-specified network statistics in STERGM, the sequential parameter differences learned from the observed networks with (4) can reflect the magnitude of structural changes over time. In summary, by penalizing the sum of sequential differences between the STERGM parameters, the proposed framework focuses on capturing significant structural changes while smoothing out minor variations.

Furthermore, the term $\lambda > 0$ is a tuning parameter for the Group Fused Lasso penalty, and the term $\boldsymbol{d} \in \mathbb{R}^{\tau-1}$ is a position dependant weight (Bleakley & Vert, 2011) such that $\boldsymbol{d}_i = \sqrt{\tau/[i(\tau-i)]}$ for $i \in [1, \tau-1]$. Intuitively, the inverse of $\boldsymbol{d}_i$ assigns a greater weight to the time point that is far from the beginning and the end of a time span, as the end points are usually not of interest for change point detection.

Figure 1 gives an overview of the proposed framework. The shaded circles on the top denote the sequence of observed networks as time passes. The dashed circles in the middle denote the sequences of formation networks $\boldsymbol{y}^{+,t}$ and dissolution networks $\boldsymbol{y}^{-,t}$ recovered from the observed networks. Note that each observed network is utilized multiple times to extract useful information that emphasizes the transition between consecutive time steps. We learn the parameters denoted by the dotted circles at the bottom, while monitoring the sequential parameter differences.

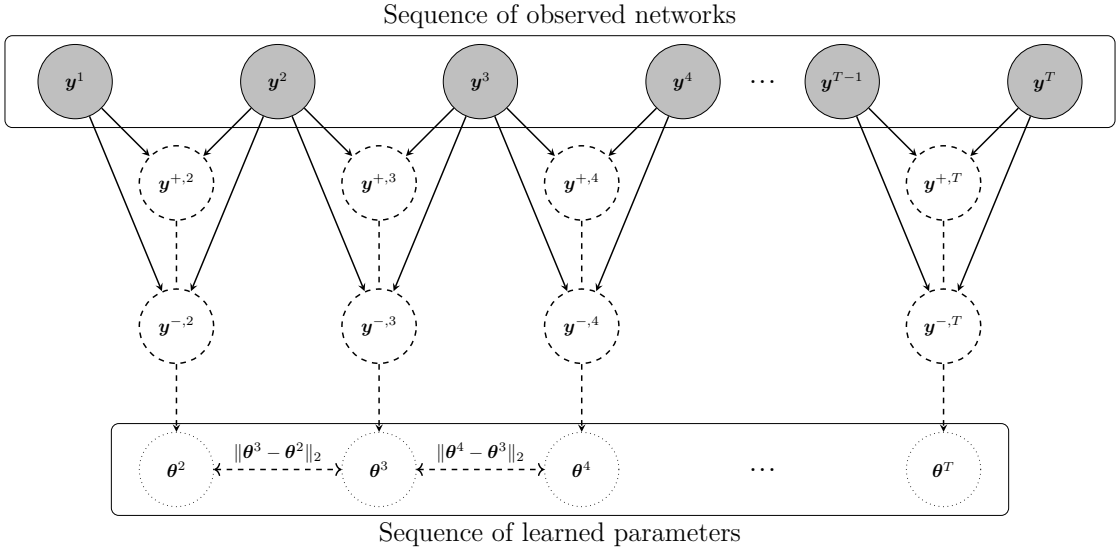

Figure 1: An illustration of change point model with STERGM.

To solve the problem in (4), we first introduce a slack variable $\boldsymbol{z} \in \mathbb{R}^{\tau \times p}$ and rewrite the original problem as a constrained optimization problem:

$$\begin{aligned} \hat{\boldsymbol{\theta}} = &\underset{\boldsymbol{\theta}}{\arg\min}\ -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{z}_{i+1,.} - \boldsymbol{z}_{i,.}\|_2}{\boldsymbol{d}_i} \\ &\text{subject to } \boldsymbol{\theta} = \boldsymbol{z}. \end{aligned} \tag{5}$$

Let $\boldsymbol{u} \in \mathbb{R}^{\tau \times p}$ be the scaled dual variable. The augmented Lagrangian can be expressed as

$$\mathcal{L}_\alpha(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u}) = -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}\|_2}{\boldsymbol{d}_i} + \frac{\alpha}{2}\|\boldsymbol{\theta} - \boldsymbol{z} + \boldsymbol{u}\|_F^2 - \frac{\alpha}{2}\|\boldsymbol{u}\|_F^2. \tag{6}$$

Levy-leduc & Harchaoui (2007) formulated a one-dimensional change point detection problem as a Lasso regression problem. Following Bleakley & Vert (2011), we make the change of variables $(\boldsymbol{\gamma}, \boldsymbol{\beta}) \in \mathbb{R}^{1\times p} \times \mathbb{R}^{(\tau-1)\times p}$ to formulate the augmented Lagrangian in (6) as a Group Lasso regression problem (Yuan & Lin, 2006; Alaíz et al., 2013), where

$$\boldsymbol{\gamma} = \boldsymbol{z}_{1,\cdot} \quad \text{and} \quad \boldsymbol{\beta}_{i,\cdot} = \frac{\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}}{\boldsymbol{d}_i} \quad \forall i \in [1, \tau - 1]. \tag{7}$$

Reversely, the matrix $\boldsymbol{z} \in \mathbb{R}^{\tau \times p}$ can also be collected by

$$\boldsymbol{z} = \mathbf{1}_{\tau,1}\boldsymbol{\gamma} + \boldsymbol{X}\boldsymbol{\beta}$$

where $\boldsymbol{X} \in \mathbb{R}^{\tau\times(\tau-1)}$ is a designed matrix with $\boldsymbol{X}_{i,j} = \boldsymbol{d}_j$ for $i > j$ and 0 otherwise. Plugging $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ into (6), we have

$$\mathcal{L}_\alpha(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{u}) = -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \|\boldsymbol{\beta}_{i,\cdot}\|_2 + \frac{\alpha}{2}\|\boldsymbol{\theta} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{u}\|_F^2 - \frac{\alpha}{2}\|\boldsymbol{u}\|_F^2. \tag{8}$$

Thus we derive an Alternating Direction Method of Multipliers (ADMM) to solve (5). The resulting ADMM is given as:

$$\boldsymbol{\theta}^{(a+1)} = \arg\min_{\boldsymbol{\theta}} -l(\boldsymbol{\theta}) + \frac{\alpha}{2}\|\boldsymbol{\theta} - \boldsymbol{z}^{(a)} + \boldsymbol{u}^{(a)}\|_F^2, \tag{9}$$

$$\boldsymbol{\gamma}^{(a+1)}, \boldsymbol{\beta}^{(a+1)} = \arg\min_{\boldsymbol{\gamma},\boldsymbol{\beta}} \lambda \sum_{i=1}^{\tau-1} \|\boldsymbol{\beta}_{i,\cdot}\|_2 + \frac{\alpha}{2}\|\boldsymbol{\theta}^{(a+1)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{u}^{(a)}\|_F^2, \tag{10}$$

$$\boldsymbol{u}^{(a+1)} = \boldsymbol{\theta}^{(a+1)} - \boldsymbol{z}^{(a+1)} + \boldsymbol{u}^{(a)}, \tag{11}$$

where a denotes the current ADMM iteration. Note that the update (10) is equivalent to

$$\boldsymbol{z}^{(a+1)} = \arg\min_{\boldsymbol{z}} \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}\|_2}{\boldsymbol{d}_i} + \frac{\alpha}{2}\|\boldsymbol{\theta}^{(a+1)} - \boldsymbol{z} + \boldsymbol{u}^{(a)}\|_F^2,$$

except that we decompose the slack variable $\boldsymbol{z}$ to work with $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ instead. Once the update (10) is completed within an ADMM iteration, we collect $\boldsymbol{z}^{(a+1)} = \mathbf{1}_{\tau,1}\boldsymbol{\gamma}^{(a+1)} + \boldsymbol{X}\boldsymbol{\beta}^{(a+1)}$ until the next decomposition of $\boldsymbol{z}$. We recursively implement the three updates until a convergence criterion is satisfied.

By adapting the idea from Boyd et al. (2011), we have the following result for the proposed ADMM procedure:

Proposition 1. *Denote the respective primal and dual residuals at the ath ADMM iteration as*

$$r_{primal}^{(a)} = \sqrt{\frac{1}{\tau \times p}\sum_{i=1}^{\tau}\sum_{j=1}^{p}(\boldsymbol{\theta}_{ij}^{(a)} - \boldsymbol{z}_{ij}^{(a)})^2} \quad \textit{and} \quad r_{dual}^{(a)} = \sqrt{\frac{1}{\tau \times p}\sum_{i=1}^{\tau}\sum_{j=1}^{p}(\boldsymbol{z}_{ij}^{(a)} - \boldsymbol{z}_{ij}^{(a-1)})^2}.$$

Assume the updates (9) and (10) attain minimum at each ADMM iteration. Then the primal residual $r_{primal}^{(a)} \to 0$ and dual residual $r_{dual}^{(a)} \to 0$ as $a \to \infty$.

The proof is provided in Appendix A. Next, we discuss the updates (9) and (10) in detail.

3.2 Updating $\boldsymbol{\theta}$

In this section, we derive the Newton-Raphson method for learning $\boldsymbol{\theta}$ in the update (9). We choose to use the Newton-Raphson method because it is more efficient than gradient descent: the Newton-Raphson method utilizes second-order information to adaptively determine the step size, leading to quadratic convergence and more stable updates (Galántai, 2000). To implement the Newton-Raphson method in a compact form, we first vectorize $\boldsymbol{\theta} \in \mathbb{R}^{\tau \times p}$ as $\vec{\boldsymbol{\theta}} = \text{vec}_{\tau p}(\boldsymbol{\theta}) \in \mathbb{R}^{\tau p \times 1}$, and we construct

$$\boldsymbol{\Delta}^t = \begin{pmatrix} \boldsymbol{\Delta}^{+,t} & \\ & \boldsymbol{\Delta}^{-,t} \end{pmatrix} \text{ and } \boldsymbol{H} = \begin{pmatrix} \boldsymbol{\Delta}^2 & & \\ & \ddots & \\ & & \boldsymbol{\Delta}^T \end{pmatrix}.$$

The matrices $\boldsymbol{\Delta}^{+,t} \in \mathbb{R}^{E \times p_1}$ and $\boldsymbol{\Delta}^{-,t} \in \mathbb{R}^{E \times p_2}$ abbreviate the respective change statistics $\Delta \boldsymbol{g}^+_{ij}(\boldsymbol{y}^{+,t})$ and $\Delta \boldsymbol{g}^-_{ij}(\boldsymbol{y}^{-,t})$ that are ordered by the dyads. The dimension of $\boldsymbol{\Delta}^t$ is thus $2E \times p$. The quantity $E = n \times n$ is due to vectorization, and the double in the number of rows is due to the separability of STERGM. In practice, the matrix $\boldsymbol{H} \in \mathbb{R}^{2\tau E \times \tau p}$ that consists of the change statistics for $t = 2, \ldots, T$ is calculated before the implementation of ADMM.

For the Hessian matrix, we also need to calculate $\vec{\boldsymbol{\mu}} = h(\boldsymbol{H} \cdot \vec{\boldsymbol{\theta}}) \in \mathbb{R}^{2\tau E \times 1}$ where $h(x) = 1/(1 + \exp(-x))$ is the element-wise sigmoid function with $h'(x) = h(x)(1 - h(x))$. Furthermore, we construct

$$\boldsymbol{W}^t = \begin{pmatrix} \boldsymbol{W}^{+,t} & \\ & \boldsymbol{W}^{-,t} \end{pmatrix} \text{ and } \boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}^2 & & \\ & \ddots & \\ & & \boldsymbol{W}^T \end{pmatrix}$$

where $\boldsymbol{W}^{+,t} = \text{diag}(\boldsymbol{\mu}^{+,t}_{ij}(1 - \boldsymbol{\mu}^{+,t}_{ij})) \in \mathbb{R}^{E \times E}$ with $\boldsymbol{\mu}^{+,t}_{ij} = h(\boldsymbol{\theta}^{+,t} \cdot \Delta \boldsymbol{g}^+_{ij}(\boldsymbol{y}^{+,t})) \in (0, 1)$. The matrix $\boldsymbol{W}^{-,t} \in \mathbb{R}^{E \times E}$ is defined similarly except for notational difference.

Proposition 2. *Using the Newton-Raphson method, the parameter $\vec{\boldsymbol{\theta}} \in \mathbb{R}^{\tau p \times 1}$ of our change point model can be updated iteratively by applying the following:*

$$\vec{\boldsymbol{\theta}}_{c+1} = \vec{\boldsymbol{\theta}}_c - \big(\boldsymbol{H}^\top \boldsymbol{W} \boldsymbol{H} + \alpha \boldsymbol{I}_{\tau p}\big)^{-1} \cdot \big(- \boldsymbol{H}^\top (\vec{\boldsymbol{y}} - \vec{\boldsymbol{\mu}}) + \alpha (\vec{\boldsymbol{\theta}}_c - \vec{\boldsymbol{z}}^{(a)} + \vec{\boldsymbol{u}}^{(a)})\big) \tag{12}$$

where c denotes the current Newton-Raphson iteration.

The derivations are provided in Appendix B. The diagonal matrix $\boldsymbol{W}$ with diagonal entries between 0 and 1, along with a quadratic form of the matrix $\boldsymbol{H}$, shows that the matrix $\boldsymbol{H}^\top \boldsymbol{W} \boldsymbol{H}$ is positive semi-definite. The identity matrix $\boldsymbol{I}_{\tau p}$ inherited from the augmentation term $\|\boldsymbol{\theta} - \boldsymbol{z} + \boldsymbol{u}\|^2_F$ in (9) ensures the Hessian matrix $\boldsymbol{H}^\top \boldsymbol{W} \boldsymbol{H} + \alpha \boldsymbol{I}_{\tau p}$ is not only invertible but also positive definite. Thus the objective function in (9) is strongly convex with respect to the parameter $\boldsymbol{\theta}$ and a global minimum is guaranteed to exist. The constructed formation and dissolution network data is vectorized in the form of $\vec{\boldsymbol{y}} \in \{0, 1\}^{2\tau E \times 1}$ to align with the dyad order of the matrix $\boldsymbol{H} \in \mathbb{R}^{2\tau E \times \tau p}$. Once the Newton-Raphson method is concluded within an ADMM iteration, we fold the updated vector $\vec{\boldsymbol{\theta}}$ back into a matrix as $\boldsymbol{\theta}^{(a+1)} = \text{vec}^{-1}_{\tau,p}(\vec{\boldsymbol{\theta}}) \in \mathbb{R}^{\tau \times p}$ before implementing the update in (10), which is discussed next.

3.3 Updating $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$

In this section, we derive the update in (10), which is equivalent to solving a Group Lasso problem. We decompose the matrix $\boldsymbol{z}$ to work with $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ instead, and the objective function is convex with respect to these parameters. With ADMM, the updates on $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ do not require the network data and the change statistics, but the updates primarily rely on the $\boldsymbol{\theta}$ learned from the update (9).

By adapting the derivation from Vert & Bleakley (2010) and Bleakley & Vert (2011), the matrix $\boldsymbol{\beta} \in \mathbb{R}^{(\tau-1) \times p}$ can be updated in a block coordinate descent manner.

Proposition 3. *We iteratively apply the following equation to update $\boldsymbol{\beta}_{i,\cdot}$ for each block $i = 1, \ldots, \tau - 1$:*

$$\boldsymbol{\beta}_{i,\cdot} \leftarrow \frac{1}{\alpha \boldsymbol{X}^\top_{\cdot,i} \boldsymbol{X}_{\cdot,i}} \Big(1 - \frac{\lambda}{\|\boldsymbol{s}_i\|_2}\Big)_+ \boldsymbol{s}_i \tag{13}$$

where $(\cdot)_+ = \max(\cdot, 0)$ *and*

$$\boldsymbol{s}_i = \alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-i}\boldsymbol{\beta}_{-i,\cdot}).$$

The derivations are provided in Appendix C, and the convergence of the procedure is monitored by the Karush-Kuhn-Tucker (KKT) conditions:

$$\lambda \frac{\boldsymbol{\beta}_{i,\cdot}}{\|\boldsymbol{\beta}_{i,\cdot}\|_2} - \alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta}) = \mathbf{0} \quad \forall \boldsymbol{\beta}_{i,\cdot} \neq \mathbf{0},$$
$$\|-\alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta})\|_2 \leq \lambda \quad \forall \boldsymbol{\beta}_{i,\cdot} = \mathbf{0}.$$

Subsequently, for any $\boldsymbol{\beta} \in \mathbb{R}^{(\tau-1)\times p}$, the minimum in $\boldsymbol{\gamma} \in \mathbb{R}^{1\times p}$ is achieved at

$$\boldsymbol{\gamma} = (1/\tau)\mathbf{1}_{1,\tau} \cdot (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \boldsymbol{X}\boldsymbol{\beta}).$$

Once the update (10) is concluded within an ADMM iteration, we collect $\boldsymbol{z} = \mathbf{1}_{\tau,1}\boldsymbol{\gamma} + \boldsymbol{X}\boldsymbol{\beta}$ and proceed to update the scaled dual variable $\boldsymbol{u} \in \mathbb{R}^{\tau\times p}$ with (11).

4 Change Point Localization and Model Selection

In this section, we discuss the choices of network statistics for change point detection with STERGM, followed by change point localization and model selection.

4.1 Network Statistics

As a probability distribution over dynamic networks, STERGM allows us to generate different networks that share similar structural patterns with the observed networks, by using a carefully designed MCMC sampling algorithm (Snijders, 2002; Krivitsky, 2017). Hence, in a dynamic network modeling problem with STERGM, network statistics are often chosen to signify the underlying process producing the observed networks or to capture important network effects interpreting for a research question.

In our change point detection problem with STERGM, network statistics are chosen to determine the types of structural changes that are searched for by the researchers. The R library `ergm` (Handcock et al., 2022) provides an extensive list of network statistics that boost the power of the proposed method. Since the underlying reasons that result in edge formation are usually different from those that result in edge dissolution, the choices of network statistics in the formation model can be different from those in the dissolution model. For an in-depth discussion of network statistics in an ERGM framework, see Handcock et al. (2003), Hunter & Handcock (2006), Snijders et al. (2006), Hunter et al. (2008a), Morris et al. (2008), Robins et al. (2009), and Blackburn & Handcock (2022).

4.2 Data-driven Threshold

Intuitively, the location of a change point is the time step where the parameter of STERGM at time t differs from that at time $t-1$. To this end, we can calculate the parameter difference between consecutive time points in $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{\tau\times p}$ as

$$\Delta\hat{\boldsymbol{\theta}}_i = \|\hat{\boldsymbol{\theta}}_{i+1,\cdot} - \hat{\boldsymbol{\theta}}_{i,\cdot}\|_2 \quad \forall i \in [1, \tau-1]$$

and declare a change point when a parameter difference is greater than a threshold.

Though researchers can choose an arbitrary threshold for $\Delta\hat{\boldsymbol{\theta}}$ based on the sensitivity of the detection, in this work we provide a data-driven threshold with the following procedures. First we standardize the parameter differences $\Delta\hat{\boldsymbol{\theta}}$ as

$$\Delta\hat{\boldsymbol{\zeta}}_i = \frac{\Delta\hat{\boldsymbol{\theta}}_i - \text{median}(\Delta\hat{\boldsymbol{\theta}})}{\text{sd}(\Delta\hat{\boldsymbol{\theta}})} \quad \forall i \in [1, \tau-1]. \tag{14}$$

Then the threshold based on the parameters learned from the data is constructed as

$$\epsilon_{\text{thr}} = \text{mean}(\Delta\hat{\boldsymbol{\zeta}}) + \mathcal{Z}_{1-\alpha} \times \text{sd}(\Delta\hat{\boldsymbol{\zeta}}) \tag{15}$$

where $\mathcal{Z}_{1-\alpha}$ is the $(1-\alpha)\%$ quantile of the standard Normal distribution. We declare a change point B_k when $\Delta\hat{\boldsymbol{\zeta}}_{B_k} > \epsilon_{\text{thr}}$. The data-driven threshold in (15) is intuitive, as the standardized parameter differences at the change points are greater than those values in between the change points, derived from the Group Fused Lasso penalty. When tracing in a plot over time, the values $\Delta\hat{\boldsymbol{\zeta}}$ can exhibit the magnitude of structural changes, in terms of the network statistics specified in the STERGM.

4.3 Model Selection

To determine the optimal set of change points over multiple STERGMs learned with different tuning parameter λ, we can use Bayesian information criterion (BIC) to perform model selection. Consider the STERGM with learned $\hat{\boldsymbol{\theta}}$ and fixed λ, we have

$$\text{BIC}(\hat{\boldsymbol{\theta}}, \lambda) = -2l(\hat{\boldsymbol{\theta}}) + \log(TN_{\text{net}}) \times p \times \text{Seg}(\hat{\boldsymbol{\theta}}, \lambda). \tag{16}$$

For a list of λ, we choose the set of change points obtained from the STERGM with the lowest BIC value.

Different from the number of nodes n, the network size N_{net} is $\binom{n}{2}$ for an undirected network and $2 \times \binom{n}{2}$ for a directed network. In general, for a dyadic dependent network, the effective network size is often smaller than N_{net} and it may be difficult to quantify the effective size (Hunter et al., 2008a). In a node clustering problem for a static network, Handcock et al. (2007) used the number of observed edges as N_{net} to quantify the effective network size. In this work, we use N_{net} to consider a greater value for the network size, since the procedure is to select a model with the lowest BIC value. Furthermore, the term $\text{Seg}(\hat{\boldsymbol{\theta}}, \lambda)$ in (16) gives the number of segments between change points $\{\hat{B}_k\}_{k=0}^{K+1}$ that are learned with a λ. In other words, $\text{Seg}(\hat{\boldsymbol{\theta}}, \lambda) = K + 1$, where K is the number of detected change points.

5 Simulated and Real Data Experiments

In this section, we evaluate the proposed method on both simulated and real data. For simulated data, we use the following three metrics to compare the performance of the proposed and competing methods. The first metric is the absolute error $|\hat{K} - K|$ where $\hat{K}$ and K are the numbers of detected and true change points, respectively. The second metric is the one-sided Hausdorff distance defined as

$$d(\hat{\mathcal{C}}|\mathcal{C}) = \max_{c \in \mathcal{C}} \min_{\hat{c} \in \hat{\mathcal{C}}} |\hat{c} - c|,$$

where $\hat{\mathcal{C}}$ and $\mathcal{C}$ are the respective sets of detected and true change points. We also report the metric $d(\mathcal{C}|\hat{\mathcal{C}})$. When $\hat{\mathcal{C}} = \emptyset$, we define $d(\hat{\mathcal{C}}|\mathcal{C}) = \infty$ and $d(\mathcal{C}|\hat{\mathcal{C}}) = -\infty$. The third metric described in van den Burg & Williams (2020) is the coverage of a partition $\mathcal{G}$ by another partition $\mathcal{G}'$, defined as

$$C(\mathcal{G}, \mathcal{G}') = \frac{1}{T} \sum_{\mathcal{A} \in \mathcal{G}} |\mathcal{A}| \cdot \max_{\mathcal{A}' \in \mathcal{G}'} \frac{|\mathcal{A} \cap \mathcal{A}'|}{|\mathcal{A} \cup \mathcal{A}'|}$$

with $\mathcal{A}, \mathcal{A}' \subseteq [1, T]$. The $\mathcal{G}$ and $\mathcal{G}'$ are collections of intervals between consecutive change points for the respective true and detected change points. Throughout, the network statistics are calculated directly from the R library `ergm` (Handcock et al., 2022) and the formulations are provided in Appendix E.

5.1 Simulations

We simulate dynamic networks from two particular models to imitate realistic social patterns. We use the Stochastic Block Model (SBM) to attain that participants with similar attributes tend to form communities, and we impose a time-dependent mechanism in the generation process. Also, we simulate dynamic networks from STERGM, which separately takes into account how relations form and dissolve over time, as their underlying social reasons are usually different.

For each specification, we provide 10 Monte Carlo simulations of dynamic networks. We let the time span $T = 100$ and the number of nodes $n = \{50, 100, 500\}$. The true change points are located at $t = \{26, 51, 76\}$,

so $K = 3$. The $K + 1 = 4$ intervals in the partition $\mathcal{G}$ are $\mathcal{A}_1 = \{1, \ldots, 25\}$, $\mathcal{A}_2 = \{26, \ldots, 50\}$, $\mathcal{A}_3 = \{51, \ldots, 75\}$, and $\mathcal{A}_4 = \{76, \ldots, 100\}$. In each specification, we report the means over 10 Monte Carlo trials for different evaluation metrics.

To detect change points with our method, we initialize the penalty parameter $\alpha = 10$. We let the tuning parameter $\lambda = 10^b$ with $b \in \{-2, -1, \ldots, 6, 7\}$. For each λ, we run $A = 200$ iterations of ADMM and the stopping criterion in (39) uses $\epsilon_{\text{tol}} = 10^{-7}$. Within each ADMM iteration, we run $C = 20$ iterations of the Newton-Raphson method, and $D = 20$ iterations for Group Lasso. The stopping criteria for the Newton-Raphson method is $\|\vec{\boldsymbol{\theta}}_{c+1} - \vec{\boldsymbol{\theta}}_c\|_2 < 10^{-3}$. To construct the data-driven threshold in (15), we use the 90% quantile of the standard Normal distribution.

Two competitor methods, gSeg (Chen & Zhang, 2015) and kerSeg (Song & Chen, 2022b) that are available in the respective R libraries `gSeg` (Chen et al., 2020b) and `kerSeg` (Song & Chen, 2022a), are provided for comparison. We use networks (nets.) and network statistics (stats.) as two types of input data to the competing methods. For gSeg, we use the minimum spanning tree to construct the similarity graph, and we use the approximated p-value of the original edge-count scan statistic. The significance level is set to $\alpha = 0.05$. For kerSeg, we use the approximated p-value of the fGKCP$_1$ and we set the significance level $\alpha = 0.001$. Throughout, we remain on these settings, since they produce good performance on average for the competitors. Changing the above settings can improve their performance on some specifications, while severely jeopardizing their performance on other specifications.

Scenario 1: Stochastic Block Model (SBM)

As in Madrid Padilla et al. (2022), we construct two probability matrices $\boldsymbol{P}, \boldsymbol{Q} \in [0,1]^{n \times n}$ and they are defined as

$$\boldsymbol{P}_{ij} = \begin{cases} 0.5, & i, j \in \mathcal{B}_l, l \in [3], \\ 0.3, & \text{otherwise}, \end{cases} \quad \text{and} \quad \boldsymbol{Q}_{ij} = \begin{cases} 0.45, & i, j \in \mathcal{B}_l, l \in [3], \\ 0.2, & \text{otherwise}, \end{cases}$$

where $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ are evenly sized clusters that form a partition of $\{1, \ldots, n\}$. We then construct a sequence of matrices $\boldsymbol{E}^t$ for $t = 1, \ldots, T$ such that

$$\boldsymbol{E}^t_{ij} = \begin{cases} \boldsymbol{P}_{ij}, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ \boldsymbol{Q}_{ij}, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

Lastly, the networks are generated with $\rho \in \{0.0, 0.5, 0.9\}$ as a time-dependent mechanism. For any ρ and $t = 1, \ldots, T-1$, we let $\boldsymbol{y}^1_{ij} \sim \text{Bernoulli}(\boldsymbol{E}^1_{ij})$ and

$$\boldsymbol{y}^{t+1}_{ij} \sim \begin{cases} \text{Bernoulli}\big(\rho(1 - \boldsymbol{E}^{t+1}_{ij}) + \boldsymbol{E}^{t+1}_{ij}\big), & \boldsymbol{y}^t_{ij} = 1, \\ \text{Bernoulli}\big((1-\rho)\boldsymbol{E}^{t+1}_{ij}\big), & \boldsymbol{y}^t_{ij} = 0. \end{cases}$$

When $\rho = 0$, the probability to draw an edge for dyad (i, j) at time $t + 1$ remains the same. This imposes a time-independent condition for a sequence of generated networks. On the contrary, when $\rho > 0$, the probability to draw an edge for dyad (i, j) becomes greater at time $t + 1$ when there exists an edge at time t, and the probability becomes smaller when there does not exist an edge at time t.

Figure 2 exhibits examples of generated networks at particular time points. Visually, Scenario 1 produces adjacency matrices with block structures, and mutuality is an important pattern in these networks. To detect the change points with our method, we use two network statistics, edge count and mutuality, in both formation and dissolution models. In the competitor methods, besides the dynamic networks $\{\boldsymbol{y}^t\}_{t=1}^T$, we also use the edge count and mutuality in $\{\boldsymbol{g}(\boldsymbol{y}^t)\}_{t=1}^T$ as another specification. Tables 1, 2, and 3 display the means of evaluation metrics for different specifications.

As expected, the kerSeg method can achieve a good performance on the covering metric $C(\mathcal{G}, \mathcal{G}')$ when $\rho = 0$, since the time-independent setting aligns with the assumption of the kerSeg method. However, the performances of gSeg and kerSeg methods are worsened when $\rho > 0$. In particular, when the networks in the sequence are time-dependent, both gSeg and kerSeg methods can effectively detect the true change points,

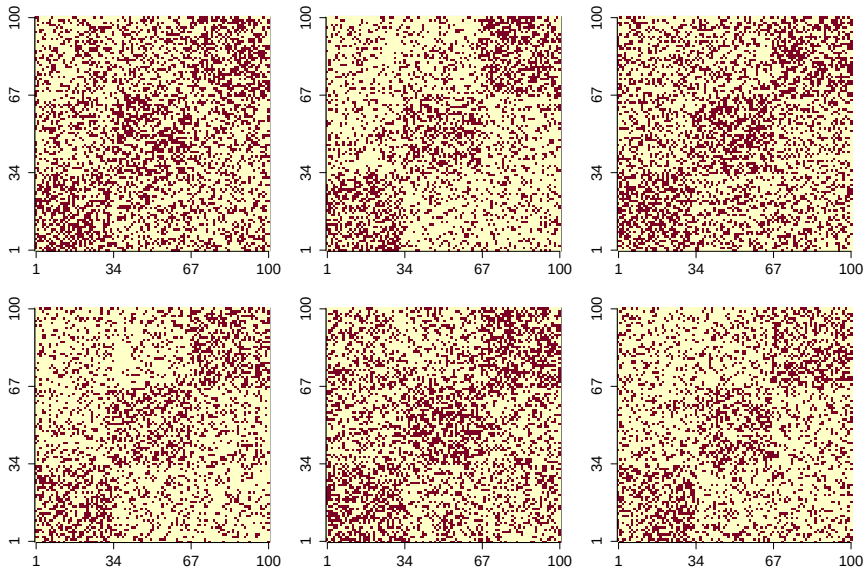

Figure 2: Examples of adjacency matrices generated from SBM with $\rho = 0.5$ and $n = 100$. In the first row, from left to right, each plot corresponds to the network at $t = 25, 50, 75$ respectively. In the second row, from left to right, each plot corresponds to the network at $t = 26, 51, 76$ respectively (the change points).

Table 1: Means of evaluation metrics for dynamic networks simulated from the Stochastic Block Model with $\rho = 0.0$. The best coverage metric is bolded.

ρ	n	Method	$\lvert\hat{K} - K\rvert \downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C}) \downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}}) \downarrow$	$C(\mathcal{G}, \mathcal{G}') \uparrow$
0.0	50	CPDstergm	0.3	0.8	2.2	95.35%
		gSeg (nets.)	2.9	Inf	$-$Inf	4.55%
		kerSeg (nets.)	0	0	0	**100**%
		gSeg (stats.)	2.1	Inf	$-$Inf	43.68%
		kerSeg (stats.)	0.1	0	0.3	99.70%
0.0	100	CPDstergm	1	0.8	5.8	89.07%
		gSeg (nets.)	2.9	Inf	$-$Inf	4.79%
		kerSeg (nets.)	0	0	0	**100**%
		gSeg (stats.)	1.9	Inf	$-$Inf	44.38%
		kerSeg (stats.)	0	0	0	**100**%
0.0	500	CPDstergm	**0**	1	1	97.07%
		gSeg (nets.)	3	Inf	$-$Inf	0%
		kerSeg (nets.)	0	0	0	**100**%
		gSeg (stats.)	2.1	Inf	$-$Inf	40.12%
		kerSeg (stats.)	0	0	0	**100**%

as the one-sided Hausdorff distance $d(\hat{\mathcal{C}}|\mathcal{C})$ are close to zeros. Yet the reversed one-sided Hausdorff distance $d(\mathcal{C}|\hat{\mathcal{C}})$ and the absolute error $|\hat{K} - K|$ show that both gSeg and kerSeg methods tend to detect excessive number of change points as the sequences of networks become noisier under the time-dependent condition.

Table 2: Means of evaluation metrics for dynamic networks simulated from the Stochastic Block Model with $\rho = 0.5$. The best coverage metric is bolded.

ρ	n	Method	$\lvert\hat{K} - K\rvert \downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C}) \downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}}) \downarrow$	$C(\mathcal{G}, \mathcal{G}') \uparrow$
0.5	50	CPDstergm	0.1	1	2.4	**97.04**%
		gSeg (nets.)	12.9	0	19.4	27.20%
		kerSeg (nets.)	6.4	0	16.6	45.50%
		gSeg (stats.)	1.8	36.6	5.8	56.04%
		kerSeg (stats.)	0.7	0	4.4	94.60%
0.5	100	CPDstergm	0	1	1	**98.04**%
		gSeg (nets.)	12.3	0	19	27.80%
		kerSeg (nets.)	6	0	15.2	47.00%
		gSeg (stats.)	1.6	Inf	$-$Inf	53.50%
		kerSeg (stats.)	0.9	0	10	92.70%
0.5	500	CPDstergm	0	1	1	**98.04**%
		gSeg (nets.)	12.3	0	19.2	27.80%
		kerSeg (nets.)	4	0	12.7	52.20%
		gSeg (stats.)	1.7	36.6	3.9	58.59%
		kerSeg (stats.)	1.3	0	7.4	91.40%

Table 3: Means of evaluation metrics for dynamic networks simulated from the Stochastic Block Model with $\rho = 0.9$. The best coverage metric is bolded.

ρ	n	Method	$\lvert\hat{K} - K\rvert \downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C}) \downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}}) \downarrow$	$C(\mathcal{G}, \mathcal{G}') \uparrow$
0.9	50	CPDstergm	0	1	1	**98.04**%
		gSeg (nets.)	12.6	0	19.2	27.50%
		kerSeg (nets.)	11	0	18.7	32.00%
		gSeg (stats.)	6.7	5.4	16.9	58.45%
		kerSeg (stats.)	4.4	0	14	70.80%
0.9	100	CPDstergm	0	1	1	**98.04**%
		gSeg (nets.)	12.6	0	19	27.50%
		kerSeg (nets.)	12	0	19	28.00%
		gSeg (stats.)	5.6	1.6	18.8	62.66%
		kerSeg (stats.)	4	0	17.3	71.50%
0.9	500	CPDstergm	0	1	1	**98.04**%
		gSeg (nets.)	12.2	0	19	27.80%
		kerSeg (nets.)	12	0	19	28.00%
		gSeg (stats.)	7.4	0.2	19.1	58.96%
		kerSeg (stats.)	5.2	0	19	66.90%

Our CPDstergm method, on average, achieves smaller absolute error, smaller one-sided Hausdorff distances, and greater coverage of interval partitions, regardless of the temporal dependence.

Another aspect worth mentioning is the usage of the network statistics in the competitor methods. The performance of gSeg and kerSeg method, in terms of the covering metric $C(\mathcal{G}, \mathcal{G}')$, improves significantly when we change the input data from networks to network statistics, which demonstrates the potential of using network level summary statistics to represent the enormous amount of individual relations.

Scenario 2: Separable Temporal ERGM

In this scenario, we employ time-homogeneous STERGMs (Krivitsky & Handcock, 2014) between change points to generate sequences of dynamic networks, using the R package `tergm` (Krivitsky & Handcock, 2022). For the following three specifications, we gradually increase the complexity of the network patterns, by adding more network statistics in the data generating process. First we use two network statistics, edge count and mutuality, in both formation and dissolution models to let $p_{\text{sim}} = 4$. The parameters are

$$\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t} = \begin{cases} -1,\ -2,\ -1,\ -2, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ -1,\ 1,\ -1,\ -1, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

Next, we include the number of triangles in both formation and dissolution models to let $p_{\text{sim}} = 6$. The parameters are

$$\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t} = \begin{cases} -2,\ 2,\ -2,\ -1,\ 2,\ 1, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ -1.5,\ 1,\ -1,\ 2,\ 1,\ 1.5, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

Finally, we include the homophily for gender, an attribute assigned to each node, in both formation and dissolution models to let $p_{\text{sim}} = 8$. The parameters are

$$\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t} = \begin{cases} -2,\ 2,\ -2,\ -1,\ -1,\ 2,\ 1,\ 1, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ -1.5,\ 1,\ -1,\ 1,\ 2,\ 1,\ 1.5,\ 2, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

The nodal attributes, $\boldsymbol{x}_i \in \{\text{Female}, \text{Male}\}$ for $i \in [n]$, are fixed across time t in the generation process.

Figure 3 exhibits examples of generated networks at particular time points. Specifically, Scenario 2 produces adjacency matrices that are sparse, which is often the case in reality. For comparison, to detect change points with our method, we use the network statistics that generate the networks in both formation and dissolution models. For the competitor methods, besides the networks, we also use the same network statistics that generate the networks as another specification. Tables 4, 5, and 6 display the means of evaluation metrics for different specifications.

For $p_{\text{sim}} = 4$, the performance of the kerSeg method in terms of the covering metric $C(\mathcal{G}, \mathcal{G}')$ improves significantly when we change the input data from networks to network statistics. However, for $p_{\text{sim}} = 6$, both gSeg and kerSeg methods tend to detect excessive number of change points when the networks are highly dyadic dependent due to the inclusion of the triangle term. Using network statistics as input can no longer improve their performance. Our CPDstergm method, which dissects the network evolution using formation and dissolution models, can achieve a good result when the networks are both temporal and dyadic dependent. Lastly, for $p_{\text{sim}} = 8$, our method permits the inclusion of nodal attributes to facilitate the change detection, besides edge information. On average, our method produces smaller absolute error, smaller one-sided Hausdorff distances, and greater coverage of interval partitions.

5.2 MIT Cellphone Data

The Massachusetts Institute of Technology (MIT) cellphone data (Eagle & Pentland, 2006) consists of human interactions via cellphone activity, among $n = 96$ participants for a duration of $T = 232$ days. The data were taken from 2004-09-15 to 2005-05-04, which covers the winter and spring vacations in the MIT academic calendar. For participants i and j, a connected edge $\boldsymbol{y}_{ij}^t = 1$ indicates that they had made at least one phone call on day t, and $\boldsymbol{y}_{ij}^t = 0$ otherwise.

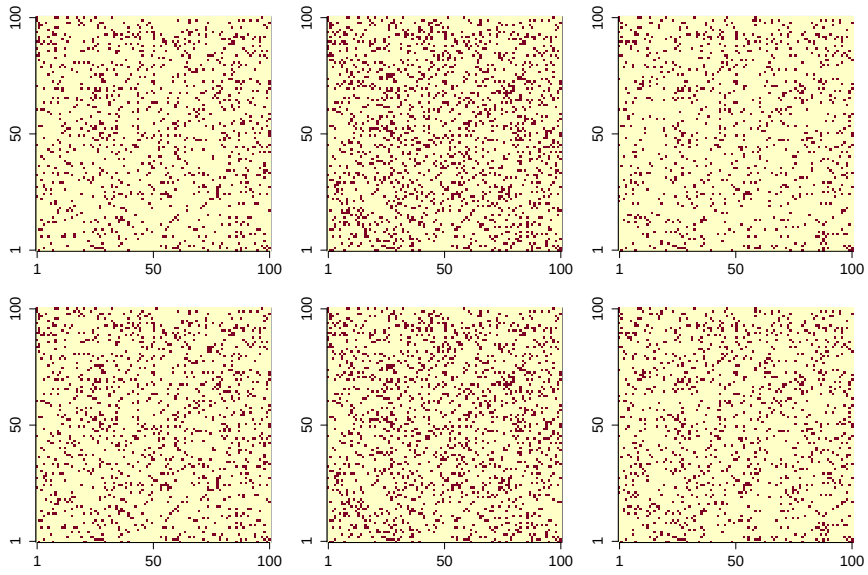

Figure 3: Examples of adjacency matrices generated from STERGM with $p_{\text{sim}} = 6$ and $n = 100$. In the first row, from left to right, each plot corresponds to the network at $t = 25, 50, 75$ respectively. In the second row, from left to right, each plot corresponds to the network at $t = 26, 51, 76$ respectively (the change points).

Table 4: Means of evaluation metrics for dynamic networks simulated from the STERGM with $p_{\text{sim}} = 4$. The best coverage metric is bolded.

p_{sim}	n	Method	$\lvert\hat{K} - K\rvert \downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C}) \downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}}) \downarrow$	$C(\mathcal{G}, \mathcal{G}') \uparrow$
4	50	CPDstergm	0	0.1	0.1	99.80%
		gSeg (nets.)	1.9	21.7	12	48.83%
		kerSeg (nets.)	2.8	0	15.8	78.30%
		gSeg (stats.)	2.1	Inf	−Inf	43.61%
		kerSeg (stats.)	0	0	0	**100**%
4	100	CPDstergm	0	0	0	**100**%
		gSeg (nets.)	1.8	18.2	17.1	44.20%
		kerSeg (nets.)	2.6	0	15.6	77.40%
		gSeg (stats.)	2.1	Inf	−Inf	30.37%
		kerSeg (stats.)	0.1	0	0.2	99.80%
4	500	CPDstergm	0	1	1	**94.96**%
		gSeg (nets.)	12	0	19	28.00%
		kerSeg (nets.)	4.6	1.7	14.4	51.65%
		gSeg (stats.)	1.9	24.9	19.8	48.41%
		kerSeg (stats.)	4.3	1.4	19.4	74.02%

As the data portrays human interactions, we use the number of (1) edges, (2) isolates, and (3) triangles to represent the occurrence of connections, the sparsity of social networks, and the transitive association of friendship, respectively. The three network statistics are used in both formation and dissolution models of

Table 5: Means of evaluation metrics for dynamic networks simulated from the STERGM with $p_{\text{sim}} = 6$. The best coverage metric is bolded.

p_{sim}	n	Method	$\lvert\hat{K} - K\rvert \downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C}) \downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}}) \downarrow$	$C(\mathcal{G}, \mathcal{G}') \uparrow$
6	50	CPDstergm	0.2	1.6	3	**91.54**%
		gSeg (nets.)	12.3	0	19	27.90%
		kerSeg (nets.)	9.7	1.4	17.9	37.62%
		gSeg (stats.)	15.8	1.5	20.1	24.55%
		kerSeg (stats.)	9.4	3.9	18	35.86%
6	100	CPDstergm	0	1	1	**94.19**%
		gSeg (nets.)	12	0	19	28.00%
		kerSeg (nets.)	9.6	1	17.5	37.66%
		gSeg (stats.)	14.9	1.9	20.3	26.13%
		kerSeg (stats.)	8	5.4	16.7	38.45%
6	500	CPDstergm	0	1	1	**98.04**%
		gSeg (nets.)	12	0	19	28.00%
		kerSeg (nets.)	8.3	0.2	16.4	42.20%
		gSeg (stats.)	1.7	45.1	4.6	49.27%
		kerSeg (stats.)	6.1	3.1	15.3	55.24%

Table 6: Means of evaluation metrics for dynamic networks simulated from the STERGM with $p_{\text{sim}} = 8$. The best coverage metric is bolded.

p_{sim}	n	Method	$\lvert\hat{K} - K\rvert \downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C}) \downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}}) \downarrow$	$C(\mathcal{G}, \mathcal{G}') \uparrow$
8	50	CPDstergm	0.4	1.7	4.4	**89.56**%
		gSeg (nets.)	13.3	0	19.6	27.20%
		kerSeg (nets.)	9.5	0.8	18.2	37.86%
		gSeg (stats.)	13.4	2.3	19.7	28.00%
		kerSeg (stats.)	8.7	4.8	18.3	36.51%
8	100	CPDstergm	0	1.6	1.6	**93.11**%
		gSeg (nets.)	12	0	19	28.00%
		kerSeg (nets.)	9.3	1.7	17.6	37.12%
		gSeg (stats.)	12.8	4.2	19.5	28.08%
		kerSeg (stats.)	8.2	5.8	18.6	36.55%
8	500	CPDstergm	0.4	12.3	2.3	**85.71**%
		gSeg (nets.)	12	0	19	28.00%
		kerSeg (nets.)	8.9	2	14.5	43.00%
		gSeg (stats.)	5.1	20.2	20.7	32.08%
		kerSeg (stats.)	9.6	2	17	37.95%

our method. For the competitors, we also use the three network statistics $\boldsymbol{g}(\boldsymbol{y}^t)$ as input data, since they provide better results than using the networks $\boldsymbol{y}^t$. Figure 4 displays $\Delta\hat{\boldsymbol{\zeta}}$ of Equation (14) and the detected change points of our method, as well as the results from the competitors. Moreover, Table 7 provides a list of potential nearby events that align with the detected change points of our method.

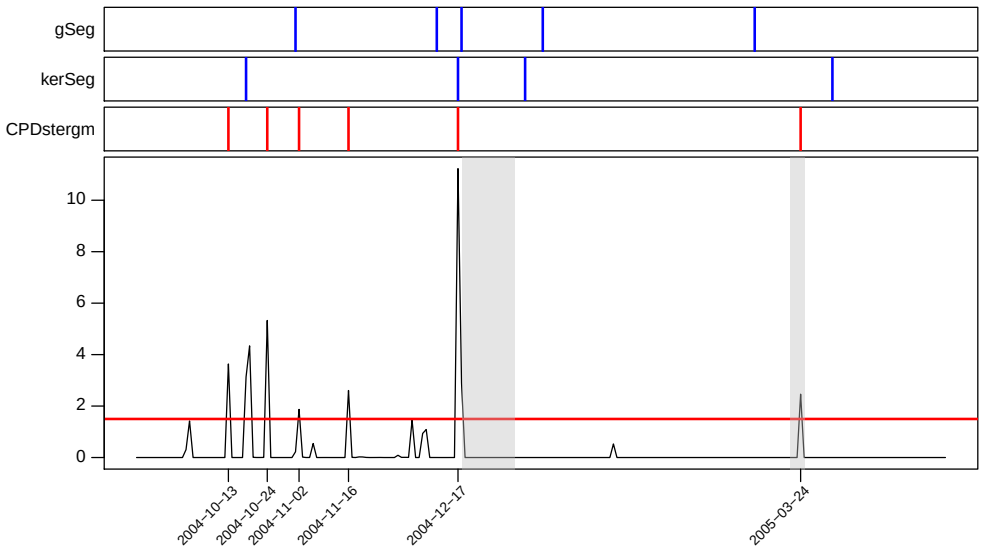

Figure 4: Visualization of $\Delta\hat{\boldsymbol{\zeta}}$ and the detected change points from our method for the MIT cellphone data. The detected change points from the competitors are also displayed. The two shaded areas correspond to the winter and spring vacations in the MIT 2004-2005 academic calendar. The data-driven threshold (red horizontal line) is calculated by (15) with $\mathcal{Z}_{0.9}$.

The two shaded areas in Figure 4 correspond to the winter and spring vacations, and our method can punctually detect the pattern change in the contact behaviors. Both gSeg and kerSeg methods can also detect the beginning of the winter vacation, but their results on the spring vacation are slightly deviated. Furthermore, we detect a few spikes in the middle of October 2004, which correspond to the annual sponsor meeting that happened on 2004-10-21. About two-thirds of the participants have prepared and attended the annual sponsor meeting, and the majority of their time has contributed to achieve project goals throughout the week (Eagle & Pentland, 2006).

Table 7: Potential nearby events that align with the detected change points (CP) of our method.

Detected CP	Potential nearby events
2004-10-13	Preparation for the Sponsor meeting
2004-10-24	2004-10-21 (Sponsor meeting)
2004-11-02	2004-11-02 (Presidential election)
2004-11-16	2004-11-17 (Last day to cancel subjects)
2004-12-17	2004-12-18 to 2005-01-02 (Winter vacation)
2005-03-24	2005-03-21 to 2005-03-25 (Spring vacation)

5.3 Stock Market Data

The stock market data consists of the weekly log returns of 29 stocks included in the Dow Jones Industrial Average (DJIA) index, and it is available in the R package `ecp` (James & Matteson, 2015). We consider the data from 2007-01-01 to 2010-01-04, which covers the 2008 worldwide economic crisis. We focus on the negative correlations among stock returns to detect the systematic anomalies in the financial market. Specifically, we first use a sliding window of width 4 to calculate the correlation matrices of the weekly log returns. We then truncate the correlation matrices by setting those entries which have negative values as 1, and the remaining as 0.

In the $T = 158$ networks, a connected edge $\boldsymbol{y}_{ij}^t = 1$ indicates the log returns of stock i and stock j are negative correlated over the four-week period that ends at week t. Moreover, the number of triangles can signify the volatility of the stock market, as the three stocks are mutually negative correlated. In general, the more triangles in a network, the more opposite movements among the stock returns, suggesting a large fluctuation in the market. On the contrary, when the number of triangles is small, the majority of the stock returns either increase or decrease at the same time, suggesting a stable trend in the market. To this end, we use the number of edges and triangles in both formation and dissolution models of our method. For the competitors, we use the networks $\boldsymbol{y}^t$ as input data, since they provide better results than using the networks statistics $\boldsymbol{g}(\boldsymbol{y}^t)$. Figure 5 displays $\Delta\hat{\boldsymbol{\zeta}}$ of Equation (14) and the detected change points of our method, as well as the results from the competitors.

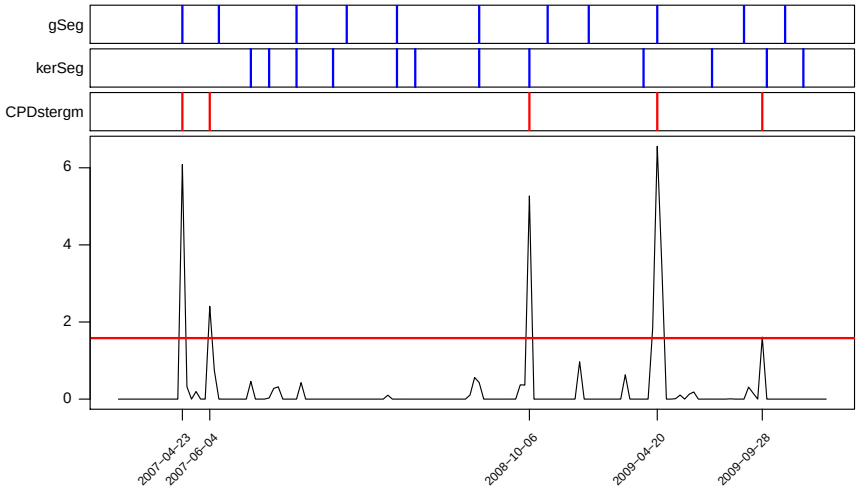

Figure 5: Visualization of $\Delta\hat{\boldsymbol{\zeta}}$ and the detected change points from our method for the stock market data. The detected change points from the competitors are also displayed. The data-driven threshold (red horizontal line) is calculated by (15) with $\mathcal{Z}_{0.9}$.

As expected, the stock market is volatile. The competitors have detected excessive number of change points, aligning with the smaller spikes in $\Delta\hat{\boldsymbol{\zeta}}$. Those change points can be detected by our method, if we manually lower the threshold to adjust the sensitivity. In this experiment, we focus on the top three spikes for real event interpretation. Table 8 presents the three detected change points and the potential nearby events. Given the networks are constructed using a sliding window, a detected change point indicates the pattern changes occur amid the four-week time span. As supporting evidence, the New Century Financial Corporation (NCFC) was the largest U.S. subprime mortgage lender in 2007, and the Lehman Brothers (LB) was one of the largest investment banks. Their bankruptcies caused by the collapse of the mortgage industry severely fueled the worldwide financial crisis, which also led the DJIA to the bottom.

Table 8: Potential nearby events that align with the top three detected change points (CP) of our method.

Detected CP	Potential nearby events
2007-04-23	2007-04-02 (NCFC filed for bankruptcy)
2008-10-06	2008-09-15 (LB filed for bankruptcy)
2009-04-20	2009-03-09 (DJIA bottomed)

6 Discussion

In this work, we study the change point detection problem in time series of graphs, which can serve as a prerequisite for dynamic network analysis. Essentially, we fit a time-heterogeneous STERGM while penalizing the sum of Euclidean norms of the parameter differences between consecutive time steps. The objective function with the Group Fused Lasso penalty is solved via Alternating Direction Method of Multipliers, and we adopt the pseudo-likelihood of STERGM to expedite parameter estimation.

The STERGM (Krivitsky & Handcock, 2014) used in our method is a flexible model to fit dynamic networks with both dyadic and temporal dependence. It manages dyad formation and dissolution separately, as the underlying reasons that induce the two processes are usually different in reality. Furthermore, the ERGM suite (Handcock et al., 2022) provides an extensive list of network statistics to capture the structural changes, and we develop an R package `CPDstergm` to implement the proposed method.

Several improvements to our change point detection method are possible. Relational phenomena by nature have degrees of strength, and dichotomizing valued networks into binary networks may introduce biases for analysis (Thomas & Blitzstein, 2011). We can extend the STERGM with a valued ERGM (Krivitsky, 2012; Desmarais & Cranmer, 2012a; Caimo & Gollini, 2020) to facilitate change point detection in dynamic valued networks. Moreover, the number of participants and their attributes are subject to change over time. It is necessary for a change point detection method to adjust the network sizes as in Krivitsky et al. (2011), and to adapt the time-evolving nodal attributes by incorporating the Exponential-family Random Network Model (ERNM) as in Fellows & Handcock (2012) and Fellows & Handcock (2013).

References

Carlos M Alaíz, Alvaro Barbero, and José R Dorronsoro. Group fused lasso. In *Artificial Neural Networks and Machine Learning–ICANN 2013: 23rd International Conference on Artificial Neural Networks Sofia, Bulgaria, September 10-13, 2013. Proceedings 23*, pp. 66–73. Springer, 2013.

Hiroyasu Ando, Akihiro Nishi, and Mark S Handcock. Statistical modeling of networked evolutionary public goods games. *arXiv preprint arXiv:2501.07007*, 2025.

Avanti Athreya, Zachary Lubberts, Youngser Park, and Carey Priebe. Euclidean mirrors and dynamics in network time series. *Journal of the American Statistical Association*, 0(0):1–41, 2024.

Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225, 1974.

Bart Blackburn and Mark S Handcock. Practical network modeling via tapered exponential-family random graph models. *Journal of Computational and Graphical Statistics*, pp. 1–14, 2022.

Kevin Bleakley and Jean-Philippe Vert. The group fused lasso for multiple change-point detection. *arXiv preprint arXiv:1106.4199*, 2011.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Tom Broekel and Marcel Bednarz. Disentangling link formation and dissolution in spatial networks: An application of a two-mode stergm to a project-based r&d network in the german biotechnology industry. *Networks and Spatial Economics*, 18:677–704, 2018.

Leland Bybee and Yves Atchadé. Change-point computation for large graphical models: A scalable algorithm for gaussian graphical models with change-points. *Journal of Machine Learning Research*, 19(11):1–38, 2018.

Alberto Caimo and Nial Friel. Bayesian inference for exponential random graph models. *Social networks*, 33 (1):41–55, 2011.

Alberto Caimo and Isabella Gollini. A multilayer exponential random graph modelling approach for weighted networks. *Computational Statistics & Data Analysis*, 142:106825, 2020.

Guodong Chen, Jesús Arroyo, Avanti Athreya, Joshua Cape, Joshua T Vogelstein, Youngser Park, Chris White, Jonathan Larson, Weiwei Yang, and Carey E Priebe. Multiple network embedding for anomaly detection in time series of graphs. *arXiv preprint arXiv:2008.10055*, 2020a.

Hao Chen. Sequential change-point detection based on nearest neighbors. *The Annals of Statistics*, 47(3): 1381–1407, 2019.

Hao Chen and Nancy Zhang. Graph-based change-point detection. *The Annals of Statistics*, 43(1):139–176, 2015.

Hao Chen, Nancy R. Zhang, Lynna Chu, and Hoseung Song. *gSeg: Graph-Based Change-Point Detection (g-Segmentation)*, 2020b. URL `https://CRAN.R-project.org/package=gSeg`. R package version 1.0.

Tianyi Chen, Zachary Lubberts, Avanti Athreya, Youngser Park, and Carey E Priebe. Euclidean mirrors and first-order changepoints in network time series. *arXiv preprint arXiv:2405.11111*, 2024.

Lynna Chu and Hao Chen. Asymptotic distribution-free change-point detection for multivariate and non-euclidean data. *The Annals of Statistics*, 47(1):382–414, 2019.

Bruce A Desmarais and Skyler J Cranmer. Statistical inference for valued-edge networks: The generalized exponential random graph model. *PloS one*, 7(1):e30136, 2012a.

Bruce A Desmarais and Skyler J Cranmer. Statistical mechanics of networks: Estimation and uncertainty. *Physica A: Statistical Mechanics and its Applications*, 391(4):1865–1876, 2012b.

Claire Donnat and Susan Holmes. Tracking network dynamics: A survey using graph distances. *The Annals of Applied Statistics*, 12(2):971–1012, 2018.

Nathan Eagle and Alex (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

Ian Fellows and Mark S. Handcock. Exponential-family random network models. *arXiv preprint arxiv:1208.0121*, 2012.

Ian E Fellows and Mark S Handcock. Analysis of partially observed networks via exponential-family random network models. *arXiv preprint arXiv:1303.1219*, 2013.

Cornelius Fritz, Emilio Dorigatti, and David Rügamer. Combining graph neural networks and spatio-temporal disease models to predict covid-19 cases in germany. *arXiv preprint arXiv:2101.00661*, 2021.

Aurel Galántai. The theory of newton's method. *Journal of Computational and Applied Mathematics*, 124 (1-2):25–44, 2000.

Charles J Geyer and Elizabeth A Thompson. Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 54(3):657–683, 1992.

Ravi Goyal and Victor De Gruttola. Dynamic network prediction. *Network Science*, 8(4):574–595, 2020.

Mark S Handcock, Garry Robins, Tom Snijders, Jim Moody, and Julian Besag. Assessing degeneracy in statistical models of social networks. Technical report, Working paper, 2003.

Mark S Handcock, Adrian E Raftery, and Jeremy M Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.

Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, Pavel N. Krivitsky, and Martina Morris. *ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks*. The Statnet Project (`https://statnet.org`), 2022. URL `https://CRAN.R-project.org/package=ergm`. R package version 4.3.2.

Steve Hanneke, Wenjie Fu, and Eric P Xing. Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4:585–605, 2010.

Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

Ruth M Hummel, David R Hunter, and Mark S Handcock. Improving simulation-based algorithms for fitting ergms. *Journal of Computational and Graphical Statistics*, 21(4):920–939, 2012.

David R Hunter and Mark S Handcock. Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583, 2006.

David R Hunter, Steven M Goodreau, and Mark S Handcock. Goodness of fit of social network models. *Journal of the American Statistical Association*, 103(481):248–258, 2008a.

David R. Hunter, Mark S. Handcock, Carter T. Butts, Steven M. Goodreau, and Martina Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3):1–29, 2008b.

Nicholas A. James and David S. Matteson. ecp: An r package for nonparametric multiple change point analysis of multivariate data. *Journal of Statistical Software*, 62(7):1–25, 2015. doi: 10.18637/jss.v062.i07. URL `https://www.jstatsoft.org/index.php/jss/article/view/v062i07`.

Binyan Jiang, Jailing Li, and Qiwei Yao. Autoregressive networks. *arXiv preprint arXiv:2010.04492*, 2020.

Yik Lun Kei, Yanzhen Chen, and Oscar Hernan Madrid Padilla. A partially separable model for dynamic valued networks. *Computational Statistics & Data Analysis*, pp. 107811, 2023.

Yik Lun Kei, Jialiang Li, Hangjian Li, Yanzhen Chen, and Oscar Hernan Madrid Padilla. Change point detection in dynamic graphs with generative model. *arXiv preprint arXiv:2404.04719*, 2024.

Mladen Kolar, Le Song, Amr Ahmed, and Eric P Xing. Estimating time-varying networks. *The Annals of Applied Statistics*, pp. 94–123, 2010.

Pavel N Krivitsky. Exponential-family random graph models for valued networks. *Electronic Journal of Statistics*, 6:1100, 2012.

Pavel N Krivitsky. Using contrastive divergence to seed monte carlo mle for exponential-family random graph models. *Computational Statistics & Data Analysis*, 107:149–161, 2017.

Pavel N Krivitsky and Mark S Handcock. A separable model for dynamic networks. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 76(1):29, 2014.

Pavel N. Krivitsky and Mark S. Handcock. *tergm: Fit, Simulate and Diagnose Models for Network Evolution Based on Exponential-Family Random Graph Models*. The Statnet Project (`https://statnet.org`), 2022. URL `https://CRAN.R-project.org/package=tergm`. R package version 4.1.0.

Pavel N Krivitsky, Mark S Handcock, and Martina Morris. Adjusting for network size and composition effects in exponential-family random graph models. *Statistical Methodology*, 8(4):319–339, 2011.

Federico Larroca, Paola Bermolen, Marcelo Fiori, and Gonzalo Mateos. Change point detection in weighted and directed random dot product graphs. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 1810–1814. IEEE, 2021.

Céline Levy-leduc and Zaïd Harchaoui. Catching change-points with lasso. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

Fuchen Liu, David Choi, Lu Xie, and Kathryn Roeder. Global spectral clustering in dynamic networks. *Proceedings of the National Academy of Sciences*, 115(5):927–932, 2018.

Matthew Ludkin, Idris Eckley, and Peter Neal. Dynamic stochastic block models: parameter estimation and detection of changes in community structure. *Statistics and Computing*, 28(6):1201–1213, 2018.

Oscar Hernan Madrid Padilla, Yi Yu, and Carey E Priebe. Change point localization in dependent dynamic nonparametric random dot product graphs. *Journal of Machine Learning Research*, 23(234):1–59, 2022.

Bernardo Marenco, Paola Bermolen, Marcelo Fiori, Federico Larroca, and Gonzalo Mateos. Online change point detection for weighted and directed random dot product graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 8:144–159, 2022.

Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(4):1119–1141, 2017.

Martina Morris, Mark S Handcock, and David R Hunter. Specification of exponential-family random graph models: terms and computational aspects. *Journal of statistical software*, 24(4):1548, 2008.

Martin Ondrus, Emily Olds, and Ivor Cribben. Factorized binary search: change point detection in the network structure of multivariate high-dimensional time series. *arXiv preprint arXiv:2103.06347*, 2021.

Marianna Pensky. Dynamic network models and graphon estimation. *The Annals of Statistics*, 47(4):2378–2403, 2019.

Garry Robins, Pip Pattison, and Peng Wang. Closure, connectivity and degree distributions: Exponential random graph (p*) models for directed social networks. *Social Networks*, 31(2):105–117, 2009.

Purnamrita Sarkar and Andrew W Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations Newsletter*, 7(2):31–40, 2005.

Daniel K Sewell and Yuguo Chen. Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512):1646–1657, 2015.

Daniel K Sewell and Yuguo Chen. Latent space models for dynamic networks with weighted edges. *Social Networks*, 44:105–116, 2016.

Tom AB Snijders. The statistical evaluation of social network dynamics. *Sociological Methodology*, 31(1): 361–395, 2001.

Tom AB Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2):1–40, 2002.

Tom AB Snijders. Models for longitudinal network data. *Models and Methods in Social Network Analysis*, 1:215–247, 2005.

Tom AB Snijders, Philippa E Pattison, Garry L Robins, and Mark S Handcock. New specifications for exponential random graph models. *Sociological methodology*, 36(1):99–153, 2006.

Tom AB Snijders, Gerhard G Van de Bunt, and Christian EG Steglich. Introduction to stochastic actor-based models for network dynamics. *Social networks*, 32(1):44–60, 2010.

Hoseung Song and Hao Chen. *kerSeg: New Kernel-Based Change-Point Detection*, 2022a. URL `https://CRAN.R-project.org/package=kerSeg`. R package version 1.0.

Hoseung Song and Hao Chen. New kernel-based change-point detection. *arXiv preprint arXiv:2206.01853*, 2022b.

David Strauss and Michael Ikeda. Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, 85(409):204–212, 1990.

Stephanie Thiemichen, Nial Friel, Alberto Caimo, and Göran Kauermann. Bayesian exponential random graph models with nodal random effects. *Social Networks*, 46:11–28, 2016.

Andrew C Thomas and Joseph K Blitzstein. Valued ties tell fewer lies: Why not to dichotomize network edges with thresholds. *arXiv preprint arXiv:1101.0788*, 2011.

Medha Uppala and Mark S. Handcock. Modeling wildfire ignition origins in southern California using linear network point processes. *The Annals of Applied Statistics*, 14(1):339 – 356, 2020.

Gerrit JJ van den Burg and Christopher KI Williams. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*, 2020.

Marijtje AJ Van Duijn, Krista J Gile, and Mark S Handcock. A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks*, 31(1):52–62, 2009.

Jean-Philippe Vert and Kevin Bleakley. Fast detection of multiple change-points shared by many signals using group lars. *Advances in Neural Information Processing Systems*, 23, 2010.

Daren Wang, Yi Yu, and Alessandro Rinaldo. Optimal change point detection and localization in sparse dynamic networks. *The Annals of Statistics*, 49(1):203–232, 2021.

Heng Wang, Minh Tang, Youngser Park, and Carey E Priebe. Locality statistics for anomaly detection in time series of graphs. *IEEE Transactions on Signal Processing*, 62(3):703–717, 2013.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Xinxun Zhang, Pengfei Jiao, Mengzhou Gao, Tianpeng Li, Yiming Wu, Huaming Wu, and Zhidong Zhao. Vggm: Variational graph gaussian mixture model for unsupervised change point detection in dynamic networks. *IEEE Transactions on Information Forensics and Security*, 2024.

Zifeng Zhao, Li Chen, and Lizhen Lin. Change-point detection in dynamic networks via graphon estimation. *arXiv preprint arXiv:1908.01823*, 2019.

A ADMM Convergence

In this section, we provide the proof for Proposition 1. Consider the constrained optimization problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}\|_2}{\boldsymbol{d}_i}$$
$$\text{subject to } \boldsymbol{\theta} - \boldsymbol{z} = \mathbf{0}.$$

The Lagrangian can be expressed as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{\rho}) = f(\boldsymbol{\theta}) + g(\boldsymbol{z}) + \text{tr}[\boldsymbol{\rho}^\top(\boldsymbol{\theta} - \boldsymbol{z})] \tag{17}$$

and the augmented Lagrangian can be expressed as

$$\mathcal{L}_\alpha(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{\rho}) = f(\boldsymbol{\theta}) + g(\boldsymbol{z}) + \text{tr}[\boldsymbol{\rho}^\top(\boldsymbol{\theta} - \boldsymbol{z})] + \frac{\alpha}{2}\|\boldsymbol{\theta} - \boldsymbol{z}\|_F^2 \tag{18}$$

where

$$f(\boldsymbol{\theta}) = -l(\boldsymbol{\theta}) \geq 0,\ g(\boldsymbol{z}) = \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}\|_2}{\boldsymbol{d}_i} \geq 0,$$

and $\boldsymbol{\rho} \in \mathbb{R}^{\tau \times p}$ is the Lagrange multiplier.

Let $(\boldsymbol{\theta}^*, \boldsymbol{z}^*, \boldsymbol{\rho}^*)$ be the optimal point of the Lagrangian $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{\rho})$ in Equation (17). Also, let

$$p^k = f(\boldsymbol{\theta}^k) + g(\boldsymbol{z}^k)$$

where k is the current ADMM iteration. Denote $\boldsymbol{r}^{k+1} = \boldsymbol{\theta}^{k+1} - \boldsymbol{z}^{k+1} \in \mathbb{R}^{\tau \times p}$ as the primal residual in the matrix form. Since $\mathcal{L}(\boldsymbol{\theta}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{\rho}^*) = p^{k+1} + \mathrm{tr}[(\boldsymbol{\rho}^*)^\top \boldsymbol{r}^{k+1}]$, and $\mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{z}^*, \boldsymbol{\rho}^*) \leq \mathcal{L}(\boldsymbol{\theta}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{\rho}^*)$, we have

$$\begin{aligned} p^* &\leq p^{k+1} + \mathrm{tr}[(\boldsymbol{\rho}^*)^\top \boldsymbol{r}^{k+1}] \\ p^* - p^{k+1} &\leq \mathrm{tr}[(\boldsymbol{\rho}^*)^\top \boldsymbol{r}^{k+1}]. \end{aligned} \tag{19}$$

Note that the dual update of ADMM procedure for the optimization problem in (17) is $\boldsymbol{\rho}^{k+1} = \boldsymbol{\rho}^k + \alpha \boldsymbol{r}^{k+1}$, so $\boldsymbol{\rho}^k = \boldsymbol{\rho}^{k+1} - \alpha \boldsymbol{r}^{k+1}$. Since $\boldsymbol{\theta}^{k+1} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_\alpha(\boldsymbol{\theta}, \boldsymbol{z}^k, \boldsymbol{\rho}^k)$, the optimal condition gives

$$\mathbf{0} \in \partial f(\boldsymbol{\theta}^{k+1}) + \boldsymbol{\rho}^k + \alpha(\boldsymbol{\theta}^{k+1} - \boldsymbol{z}^k). \tag{20}$$

Substituting the $\boldsymbol{\rho}^k$ in (20) with $\boldsymbol{\rho}^{k+1} - \alpha \boldsymbol{r}^{k+1}$, and then adding and subtracting $\alpha \boldsymbol{z}^{k+1}$ to the right hand side of (20), we have

$$\begin{aligned} \mathbf{0} &\in \partial f(\boldsymbol{\theta}^{k+1}) + (\boldsymbol{\rho}^{k+1} - \alpha \boldsymbol{r}^{k+1}) + \alpha(\boldsymbol{\theta}^{k+1} - \boldsymbol{z}^k) + \alpha \boldsymbol{z}^{k+1} - \alpha \boldsymbol{z}^{k+1} \\ \mathbf{0} &\in \partial f(\boldsymbol{\theta}^{k+1}) + \boldsymbol{\rho}^{k+1} - \alpha \boldsymbol{r}^{k+1} + \alpha\big(\boldsymbol{\theta}^{k+1} - \boldsymbol{z}^{k+1} + (\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)\big) \\ \mathbf{0} &\in \partial f(\boldsymbol{\theta}^{k+1}) + \boldsymbol{\rho}^{k+1} - \alpha \boldsymbol{r}^{k+1} + \alpha\big(\boldsymbol{r}^{k+1} + (\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)\big) \\ \mathbf{0} &\in \partial f(\boldsymbol{\theta}^{k+1}) + \boldsymbol{\rho}^{k+1} + \alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k). \end{aligned}$$

This implies that $\boldsymbol{\theta}^{k+1}$ minimizes the objective function

$$f(\boldsymbol{\theta}) + \mathrm{tr}\big[\big(\boldsymbol{\rho}^{k+1} + \alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)\big)^\top \boldsymbol{\theta}\big]$$

and

$$f(\boldsymbol{\theta}^{k+1}) + \mathrm{tr}\big[\big(\boldsymbol{\rho}^{k+1} + \alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)\big)^\top \boldsymbol{\theta}^{k+1}\big] \leq f(\boldsymbol{\theta}^*) + \mathrm{tr}\big[\big(\boldsymbol{\rho}^{k+1} + \alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)\big)^\top \boldsymbol{\theta}^*\big]. \tag{21}$$

Similarly, since $\boldsymbol{z}^{k+1} = \arg\min_{\boldsymbol{z}} \mathcal{L}_\alpha(\boldsymbol{\theta}^{k+1}, \boldsymbol{z}, \boldsymbol{\rho}^k)$, the optimal condition gives

$$\mathbf{0} \in \partial g(\boldsymbol{z}^{k+1}) - \boldsymbol{\rho}^k - \alpha(\boldsymbol{\theta}^{k+1} - \boldsymbol{z}^{k+1}). \tag{22}$$

Substituting the $\boldsymbol{\rho}^k$ in (22) with $\boldsymbol{\rho}^{k+1} - \alpha \boldsymbol{r}^{k+1}$, we have

$$\begin{aligned} \mathbf{0} &\in \partial g(\boldsymbol{z}^{k+1}) - (\boldsymbol{\rho}^{k+1} - \alpha \boldsymbol{r}^{k+1}) - \alpha \boldsymbol{r}^{k+1} \\ \mathbf{0} &\in \partial g(\boldsymbol{z}^{k+1}) - \boldsymbol{\rho}^{k+1}. \end{aligned}$$

This implies that $\boldsymbol{z}^{k+1}$ minimizes the objective function

$$g(\boldsymbol{z}) - \mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top \boldsymbol{z}]$$

and

$$g(\boldsymbol{z}^{k+1}) - \mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top \boldsymbol{z}^{k+1}] \leq g(\boldsymbol{z}^*) - \mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top \boldsymbol{z}^*]. \tag{23}$$

Adding the two Inequalities (21) and (23) together, we have

$$\begin{aligned} f(\boldsymbol{\theta}^{k+1}) + g(\boldsymbol{z}^{k+1}) - f(\boldsymbol{\theta}^*) - g(\boldsymbol{z}^*) &\leq \mathrm{tr}\big[\big(\boldsymbol{\rho}^{k+1} + \alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)\big)^\top (\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1})\big] + \mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top (\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)] \\ p^{k+1} - p^* &\leq \mathrm{tr}\big[\big(\boldsymbol{\rho}^{k+1} + \alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)\big)^\top (\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1})\big] + \mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top (\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)]. \end{aligned} \tag{24}$$

Separating and grouping the terms on the right hand side in (24), we have

$$\begin{aligned}
p^{k+1} - p^* &\leq \mathrm{tr}\big[(\boldsymbol{\rho}^{k+1})^\top(\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1})\big] + \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top(\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1})\big] + \mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top(\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)] \\
p^{k+1} - p^* &\leq \mathrm{tr}\big[(\boldsymbol{\rho}^{k+1})^\top\big(\boldsymbol{\theta}^* - \boldsymbol{z}^* - (\boldsymbol{\theta}^{k+1} - \boldsymbol{z}^{k+1})\big)\big] + \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top(\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1})\big] \\
p^{k+1} - p^* &\leq -\mathrm{tr}\big[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{r}^{k+1}\big] + \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top(\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1})\big] \qquad (25)
\end{aligned}$$

where $\boldsymbol{\theta}^* - \boldsymbol{z}^* = \mathbf{0}$ and $\boldsymbol{\theta}^{k+1} - \boldsymbol{z}^{k+1} = \boldsymbol{r}^{k+1}$. Moreover, note that

$$\begin{aligned}
\boldsymbol{r}^{k+1} &= \boldsymbol{\theta}^{k+1} - \boldsymbol{z}^{k+1} - (\boldsymbol{\theta}^* - \boldsymbol{z}^*) \\
\boldsymbol{r}^{k+1} &= (\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^*) - (\boldsymbol{z}^{k+1} - \boldsymbol{z}^*) \\
-\boldsymbol{r}^{k+1} &= (\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1}) + (\boldsymbol{z}^{k+1} - \boldsymbol{z}^*) \\
(\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1}) &= -\boldsymbol{r}^{k+1} - (\boldsymbol{z}^{k+1} - \boldsymbol{z}^*).
\end{aligned}$$

Substituting the term $(\boldsymbol{\theta}^* - \boldsymbol{\theta}^{k+1})$ in Inequality (25), we have

$$\begin{aligned}
p^{k+1} - p^* &\leq -\mathrm{tr}\big[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{r}^{k+1}\big] + \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top\big(-\boldsymbol{r}^{k+1} - (\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)\big)\big] \\
p^{k+1} - p^* &\leq -\mathrm{tr}\big[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{r}^{k+1}\big] - \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big] - \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)\big]. \qquad (26)
\end{aligned}$$

Then adding Inequalities (19) and (26), we have

$$\begin{aligned}
0 &\leq \mathrm{tr}[(\boldsymbol{\rho}^*)^\top\boldsymbol{r}^{k+1}] - \mathrm{tr}\big[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{r}^{k+1}\big] - \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big] - \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)\big] \\
0 &\leq \mathrm{tr}[(\boldsymbol{\rho}^* - \boldsymbol{\rho}^{k+1})^\top\boldsymbol{r}^{k+1}] - \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big] - \mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)\big] \\
0 &\geq 2\mathrm{tr}[(\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^*)^\top\boldsymbol{r}^{k+1}] + 2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big] + 2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)\big] \qquad (27)
\end{aligned}$$

where (27) followed as we multiply the inequality by two and change the sign on both sides.

Now we consider the expansion of the first term on the right hand side in Inequality (27). Substituting the $\boldsymbol{\rho}^{k+1}$ with the dual update $\boldsymbol{\rho}^{k+1} = \boldsymbol{\rho}^k + \alpha\boldsymbol{r}^{k+1}$, we have

$$\begin{aligned}
&2\mathrm{tr}[(\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^*)^\top\boldsymbol{r}^{k+1}] \qquad (28) \\
&= 2\mathrm{tr}[(\boldsymbol{\rho}^k + \alpha\boldsymbol{r}^{k+1} - \boldsymbol{\rho}^*)^\top\boldsymbol{r}^{k+1}] \\
&= 2\mathrm{tr}[(\boldsymbol{\rho}^k - \boldsymbol{\rho}^*)^\top\boldsymbol{r}^{k+1}] + 2\mathrm{tr}[\alpha(\boldsymbol{r}^{k+1})^\top(\boldsymbol{r}^{k+1})] \\
&= 2\mathrm{tr}[(\boldsymbol{\rho}^k - \boldsymbol{\rho}^*)^\top\boldsymbol{r}^{k+1}] + \alpha\|\boldsymbol{r}^{k+1}\|_F^2 + \alpha\|\boldsymbol{r}^{k+1}\|_F^2. \qquad (29)
\end{aligned}$$

Since $\boldsymbol{\rho}^{k+1} = \boldsymbol{\rho}^k + \alpha\boldsymbol{r}^{k+1}$, we also have $\boldsymbol{r}^{k+1} = \frac{1}{\alpha}(\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^k)$. Substituting the $\boldsymbol{r}^{k+1}$ in the first two terms of (29) and expanding the matrix multiplications, the expression in (28) proceeds as

$$\begin{aligned}
&2\mathrm{tr}[(\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^*)^\top\boldsymbol{r}^{k+1}] \\
&= \frac{1}{\alpha}\mathrm{tr}[2(\boldsymbol{\rho}^k - \boldsymbol{\rho}^*)^\top(\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^k)] + \alpha\frac{1}{\alpha^2}\mathrm{tr}[(\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^k)^\top(\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^k)] + \alpha\|\boldsymbol{r}^{k+1}\|_F^2 \\
&= \frac{1}{\alpha}\mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{\rho}^{k+1} - 2(\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^{k+1} + 2(\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^k - (\boldsymbol{\rho}^k)^\top\boldsymbol{\rho}^k] + \alpha\|\boldsymbol{r}^{k+1}\|_F^2 \\
&= \frac{1}{\alpha}\mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{\rho}^{k+1} - 2(\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^{k+1} + 2(\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^k - (\boldsymbol{\rho}^k)^\top\boldsymbol{\rho}^k + (\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^* - (\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^*] + \alpha\|\boldsymbol{r}^{k+1}\|_F^2 \\
&= \frac{1}{\alpha}\mathrm{tr}\big\{\big[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{\rho}^{k+1} - 2(\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^{k+1} + (\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^*\big] - \big[(\boldsymbol{\rho}^k)^\top\boldsymbol{\rho}^k - 2(\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^k + (\boldsymbol{\rho}^*)^\top\boldsymbol{\rho}^*\big]\big\} + \alpha\|\boldsymbol{r}^{k+1}\|_F^2 \\
&= \frac{1}{\alpha}\big(\|\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^*\|_F^2 - \|\boldsymbol{\rho}^k - \boldsymbol{\rho}^*\|_F^2\big) + \alpha\|\boldsymbol{r}^{k+1}\|_F^2. \qquad (30)
\end{aligned}$$

Next we consider the expansion of the second and third terms on the right hand side in Inequality (27), with an additional term $\alpha\|\boldsymbol{r}^{k+1}\|_F^2$. By adding and subtracting $\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2$, we have

$$\begin{aligned}
&\alpha\|\boldsymbol{r}^{k+1}\|_F^2+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big]+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\big] \qquad (31)\\
&=\alpha\|\boldsymbol{r}^{k+1}\|_F^2+\alpha\mathrm{tr}\big[2(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big]+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\big]+\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2-\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2\\
&=\alpha\Big\{\|\boldsymbol{r}^{k+1}\|_F^2+\mathrm{tr}\big[2(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big]+\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2\Big\}-\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\big]\\
&=\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2-\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\big] \qquad (32)
\end{aligned}$$

Since $\boldsymbol{z}^{k+1}-\boldsymbol{z}^*=(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)+(\boldsymbol{z}^k-\boldsymbol{z}^*)$, we substitute the $\boldsymbol{z}^{k+1}-\boldsymbol{z}^*$ in the third term of (32) so that (31) proceeds as

$$\begin{aligned}
&\alpha\|\boldsymbol{r}^{k+1}\|_F^2+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big]+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\big]\\
&=\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2-\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2+2\mathrm{tr}\Big\{\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top[(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)+(\boldsymbol{z}^k-\boldsymbol{z}^*)]\Big\}\\
&=\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2-\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2+\alpha\mathrm{tr}\Big\{(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top[2(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)+2(\boldsymbol{z}^k-\boldsymbol{z}^*)]\Big\}\\
&=\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2+\alpha\mathrm{tr}\Big\{(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top[(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)+2(\boldsymbol{z}^k-\boldsymbol{z}^*)]\Big\}. \qquad (33)
\end{aligned}$$

Since $\boldsymbol{z}^{k+1}-\boldsymbol{z}^k=(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)-(\boldsymbol{z}^k-\boldsymbol{z}^*)$, we sequentially substitute the $\boldsymbol{z}^{k+1}-\boldsymbol{z}^k$ in (33) so that (31) proceeds as

$$\begin{aligned}
&\alpha\|\boldsymbol{r}^{k+1}\|_F^2+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big]+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\big]\\
&=\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2+\alpha\mathrm{tr}\Big\{(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top[(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)-(\boldsymbol{z}^k-\boldsymbol{z}^*)+2(\boldsymbol{z}^k-\boldsymbol{z}^*)]\Big\}\\
&=\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2+\alpha\mathrm{tr}\Big\{(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top[(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)+(\boldsymbol{z}^k-\boldsymbol{z}^*)]\Big\}\\
&=\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2+\alpha\mathrm{tr}\Big\{[(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)-(\boldsymbol{z}^k-\boldsymbol{z}^*)]^\top[(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)+(\boldsymbol{z}^k-\boldsymbol{z}^*)]\Big\}\\
&=\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2+\alpha\big[\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^*\|_F^2-\|\boldsymbol{z}^k-\boldsymbol{z}^*\|_F^2\big]. \qquad (34)
\end{aligned}$$

Combining the results from (30) and (34) to substitute the corresponding terms on the right hand side of Inequality (27), we have

$$\begin{aligned}
&2\mathrm{tr}[(\boldsymbol{\rho}^{k+1}-\boldsymbol{\rho}^*)^\top\boldsymbol{r}^{k+1}]+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top\boldsymbol{r}^{k+1}\big]+2\mathrm{tr}\big[\alpha(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^*)\big]\le 0\\
&\frac{1}{\alpha}\big(\|\boldsymbol{\rho}^{k+1}-\boldsymbol{\rho}^*\|_F^2-\|\boldsymbol{\rho}^k-\boldsymbol{\rho}^*\|_F^2\big)+\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2+\alpha\big[\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^*\|_F^2-\|\boldsymbol{z}^k-\boldsymbol{z}^*\|_F^2\big]\le 0\\
&\frac{1}{\alpha}\|\boldsymbol{\rho}^{k+1}-\boldsymbol{\rho}^*\|_F^2+\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^*\|_F^2-\frac{1}{\alpha}\|\boldsymbol{\rho}^k-\boldsymbol{\rho}^*\|_F^2-\alpha\|\boldsymbol{z}^k-\boldsymbol{z}^*\|_F^2+\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2\le 0. \qquad (35)
\end{aligned}$$

Define

$$V^k=\frac{1}{\alpha}\|\boldsymbol{\rho}^k-\boldsymbol{\rho}^*\|_F^2+\alpha\|\boldsymbol{z}^k-\boldsymbol{z}^*\|_F^2\ge 0.$$

Then Inequality (35) can be expressed as

$$\begin{aligned}
&V^{k+1}-V^k\le-\alpha\|\boldsymbol{r}^{k+1}+(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)\|_F^2\\
&V^{k+1}-V^k\le-\alpha\big(\|\boldsymbol{r}^{k+1}\|_F^2+2\mathrm{tr}[(\boldsymbol{r}^{k+1})^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)]+\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2\big)\\
&V^{k+1}-V^k\le-\alpha\|\boldsymbol{r}^{k+1}\|_F^2-\alpha\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|_F^2-2\alpha\mathrm{tr}[(\boldsymbol{r}^{k+1})^\top(\boldsymbol{z}^{k+1}-\boldsymbol{z}^k)]. \qquad (36)
\end{aligned}$$

Recall that $\boldsymbol{z}^{k+1}$ minimizes $g(\boldsymbol{z})-\mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{z}]$ and $\boldsymbol{z}^k$ minimizes $g(\boldsymbol{z})-\mathrm{tr}[(\boldsymbol{\rho}^k)^\top\boldsymbol{z}]$. Then

$$g(\boldsymbol{z}^{k+1})-\mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{z}^{k+1}]\le g(\boldsymbol{z}^k)-\mathrm{tr}[(\boldsymbol{\rho}^{k+1})^\top\boldsymbol{z}^k]$$

and

$$g(\boldsymbol{z}^k) - \text{tr}[(\boldsymbol{\rho}^k)^\top \boldsymbol{z}^k] \leq g(\boldsymbol{z}^{k+1}) - \text{tr}[(\boldsymbol{\rho}^k)^\top \boldsymbol{z}^{k+1}].$$

Adding the above two inequalities, we have

$$\begin{aligned}
\text{tr}[(\boldsymbol{\rho}^k)^\top \boldsymbol{z}^{k+1}] + \text{tr}[(\boldsymbol{\rho}^{k+1})^\top \boldsymbol{z}^k] - \text{tr}[(\boldsymbol{\rho}^{k+1})^\top \boldsymbol{z}^{k+1}] - \text{tr}[(\boldsymbol{\rho}^k)^\top \boldsymbol{z}^k] &\leq 0 \\
\text{tr}[(\boldsymbol{\rho}^k)^\top (\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)] + \text{tr}[(\boldsymbol{\rho}^{k+1})^\top (\boldsymbol{z}^k - \boldsymbol{z}^{k+1})] &\leq 0 \\
-\text{tr}[(\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^k)^\top (\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)] &\leq 0.
\end{aligned}$$

Recall $\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^k = \alpha \boldsymbol{r}^{k+1}$. Then

$$-\alpha \text{tr}[(\boldsymbol{r}^{k+1})^\top (\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)] \leq 0.$$

Back to Inequality (36), we can see that

$$V^{k+1} - V^k \leq -\alpha \|\boldsymbol{r}^{k+1}\|_F^2 - \alpha \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|_F^2$$

also hold. Since $V^k \geq 0$ and

$$V^{k+1} \leq V^k - \alpha\big(\|\boldsymbol{r}^{k+1}\|_F^2 + \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|_F^2\big), \tag{37}$$

we know that V_k is a bounded by below decreasing sequence.

By iterating (37), we have

$$\alpha \sum_{k=0}^{\infty} \big(\|\boldsymbol{r}^{k+1}\|_F^2 + \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|_F^2\big) \leq V^0$$

which implies the primal residual $\|\boldsymbol{r}^{k+1}\|_F^2 = \|\boldsymbol{\theta}^{k+1} - \boldsymbol{z}^{k+1}\|_F^2 \to 0$ and dual residual $\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|_F^2 \to 0$ as $k \to \infty$. Similarly, for

$$r^k_{\text{primal}} = \sqrt{\frac{1}{\tau \times p} \sum_{i=1}^{\tau} \sum_{j=1}^{p} (\boldsymbol{\theta}^k_{ij} - \boldsymbol{z}^k_{ij})^2} \quad \text{and} \quad r^k_{\text{dual}} = \sqrt{\frac{1}{\tau \times p} \sum_{i=1}^{\tau} \sum_{j=1}^{p} (\boldsymbol{z}^k_{ij} - \boldsymbol{z}^{k-1}_{ij})^2},$$

we have $r^k_{\text{primal}} \to 0$ and $r^k_{\text{dual}} \to 0$ as $k \to \infty$. This concludes the proof for Proposition 1.

B Newton-Raphson Method for Updating $\boldsymbol{\theta}$

In this section, we provide the derivation for Proposition 2. Specifically, we derive the gradient and Hessian for the Newton-Raphson method to update $\boldsymbol{\theta}$. The first-order derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{+,t}$, the parameter in the formation model at a particular time point t, is

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}^{+,t}}\, l(\boldsymbol{\theta}) &= \sum_{(i,j)\in\mathbb{Y}} \Big\{ \boldsymbol{y}^{+,t}_{ij} \Delta \boldsymbol{g}^+_{ij}(\boldsymbol{y}^{+,t}) - \frac{\exp[\boldsymbol{\theta}^{+,t} \cdot \Delta \boldsymbol{g}^+_{ij}(\boldsymbol{y}^{+,t})]}{1 + \exp[\boldsymbol{\theta}^{+,t} \cdot \Delta \boldsymbol{g}^+_{ij}(\boldsymbol{y}^{+,t})]} \Delta \boldsymbol{g}^+_{ij}(\boldsymbol{y}^{+,t}) \Big\} \\
&= \sum_{(i,j)\in\mathbb{Y}} (\boldsymbol{y}^{+,t}_{ij} - \boldsymbol{\mu}^{+,t}_{ij})[\Delta \boldsymbol{g}^+_{ij}(\boldsymbol{y}^{+,t})]
\end{aligned}$$

where $\boldsymbol{\mu}^{+,t}_{ij} = h(\boldsymbol{\theta}^{+,t} \cdot \Delta \boldsymbol{g}^+_{ij}(\boldsymbol{y}^{+,t}))$. The $h(x) = 1/(1 + \exp(-x))$ is the element-wise sigmoid function. Likewise, the first-order derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{-,t}$, the parameter in the dissolution model at a particular time point t, is similar except for notational difference.

Denote the objective function in Equation (9) as $\mathcal{L}_\alpha(\boldsymbol{\theta})$. To update the parameters $\boldsymbol{\theta} \in \mathbb{R}^{\tau \times p}$ in a compact form, we first vectorize it as $\vec{\boldsymbol{\theta}} = \text{vec}_{\tau p}(\boldsymbol{\theta}) \in \mathbb{R}^{\tau p \times 1}$. The matrices $\boldsymbol{z} \in \mathbb{R}^{\tau \times p}$ and $\boldsymbol{u} \in \mathbb{R}^{\tau \times p}$ are also vectorized as $\vec{\boldsymbol{z}} = \text{vec}_{\tau p}(\boldsymbol{z}) \in \mathbb{R}^{\tau p \times 1}$ and $\vec{\boldsymbol{u}} = \text{vec}_{\tau p}(\boldsymbol{u}) \in \mathbb{R}^{\tau p \times 1}$. With the constructed matrices $\boldsymbol{H} \in \mathbb{R}^{2\tau E \times \tau p}$ and $\boldsymbol{W} \in \mathbb{R}^{2\tau E \times 2\tau E}$, the gradient of $\mathcal{L}_\alpha(\boldsymbol{\theta})$ with respect to $\vec{\boldsymbol{\theta}}$ is

$$\nabla_{\vec{\boldsymbol{\theta}}}\, \mathcal{L}_\alpha(\boldsymbol{\theta}) = -\boldsymbol{H}^\top (\vec{\boldsymbol{y}} - \vec{\boldsymbol{\mu}}) + \alpha\big(\vec{\boldsymbol{\theta}} - \vec{\boldsymbol{z}}^{(a)} + \vec{\boldsymbol{u}}^{(a)}\big)$$

where $\vec{\boldsymbol{\mu}} = h(\boldsymbol{H} \cdot \vec{\boldsymbol{\theta}}) \in \mathbb{R}^{2\tau E \times 1}$.

Furthermore, the second order derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{+,t}$ is

$$\nabla^2_{\boldsymbol{\theta}^{+,t}}\, l(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathbb{Y}} -\boldsymbol{\mu}_{ij}^{+,t}(1-\boldsymbol{\mu}_{ij}^{+,t})[\Delta \boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})\Delta \boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})^\top]$$

and the second order derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{-,t}$ is similar except for notational difference. Thus, the Hessian of $\mathcal{L}_\alpha(\boldsymbol{\theta})$ with respect to $\vec{\boldsymbol{\theta}} \in \mathbb{R}^{\tau p\times 1}$ is

$$\nabla^2_{\vec{\boldsymbol{\theta}}}\, \mathcal{L}_\alpha(\boldsymbol{\theta}) = \boldsymbol{H}^\top \boldsymbol{W}\boldsymbol{H} + \alpha \boldsymbol{I}_{\tau p}$$

where $\boldsymbol{I}_{\tau p}$ is the identity matrix. By using the Newton-Raphson method, the $\vec{\boldsymbol{\theta}} \in \mathbb{R}^{\tau p\times 1}$ is updated as

$$\vec{\boldsymbol{\theta}}_{c+1} = \vec{\boldsymbol{\theta}}_c - \big(\boldsymbol{H}^\top \boldsymbol{W}\boldsymbol{H} + \alpha \boldsymbol{I}_{\tau p}\big)^{-1} \cdot \big(-\boldsymbol{H}^\top(\vec{\boldsymbol{y}} - \vec{\boldsymbol{\mu}}) + \alpha(\vec{\boldsymbol{\theta}}_c - \vec{\boldsymbol{z}}^{(a)} + \vec{\boldsymbol{u}}^{(a)})\big)$$

where c denotes the current Newton-Raphson iteration. Note that both $\boldsymbol{W}$ and $\vec{\boldsymbol{\mu}}$ are also calculated based on $\vec{\boldsymbol{\theta}}_c$.

C Group Lasso for Updating β

In this section, we provide the derivation for Proposition 3 of learning $\boldsymbol{\beta}$, which is equivalent to solving a Group Lasso problem (Yuan & Lin, 2006). Denote the objective function in (10) as $\mathcal{L}_\alpha(\boldsymbol{\gamma}, \boldsymbol{\beta})$. When $\boldsymbol{\beta}_{i,\cdot} \neq \mathbf{0}$, the first-order derivative of $\mathcal{L}_\alpha(\boldsymbol{\gamma}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}_{i,\cdot}$ is

$$\frac{\partial}{\partial \boldsymbol{\beta}_{i,\cdot}} \mathcal{L}_\alpha(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \lambda \frac{\boldsymbol{\beta}_{i,\cdot}}{\|\boldsymbol{\beta}_{i,\cdot}\|_2} - \alpha \boldsymbol{X}_{\cdot,i}^\top(\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,i}\boldsymbol{\beta}_{i,\cdot} - \boldsymbol{X}_{\cdot,-i}\boldsymbol{\beta}_{-i,\cdot})$$

where $\boldsymbol{X}_{\cdot,i} \in \mathbb{R}^{\tau\times 1}$ is the ith column of matrix $\boldsymbol{X} \in \mathbb{R}^{\tau\times(\tau-1)}$ and $\boldsymbol{\beta}_{i,\cdot} \in \mathbb{R}^{1\times p}$ is the ith row of matrix $\boldsymbol{\beta} \in \mathbb{R}^{(\tau-1)\times p}$. Setting the gradient to $\mathbf{0}$, we have

$$\boldsymbol{\beta}_{i,\cdot} = \Big(\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i} + \frac{\lambda}{\|\boldsymbol{\beta}_{i,\cdot}\|_2}\Big)^{-1} \boldsymbol{s}_i \tag{38}$$

where

$$\boldsymbol{s}_i = \alpha \boldsymbol{X}_{\cdot,i}^\top(\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-i}\boldsymbol{\beta}_{-i,\cdot}).$$

Taking the Euclidean norm of (38) on both sides and rearrange the terms, we have

$$\|\boldsymbol{\beta}_{i,\cdot}\|_2 = (\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i})^{-1}(\|\boldsymbol{s}_i\|_2 - \lambda).$$

Plugging $\|\boldsymbol{\beta}_{i,\cdot}\|_2$ into (38), the solution of $\boldsymbol{\beta}_{i,\cdot}$ is

$$\boldsymbol{\beta}_{i,\cdot} = \frac{1}{\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i}}\Big(1 - \frac{\lambda}{\|\boldsymbol{s}_i\|_2}\Big)\boldsymbol{s}_i.$$

When $\boldsymbol{\beta}_{i,\cdot} = \mathbf{0}$, the subgradient $\boldsymbol{v}$ of $\|\boldsymbol{\beta}_{i,\cdot}\|_2$ needs to satisfy $\|\boldsymbol{v}\|_2 \leq 1$. Since

$$\mathbf{0} \in \lambda \boldsymbol{v} - \alpha \boldsymbol{X}_{\cdot,i}^\top(\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-i}\boldsymbol{\beta}_{-i,\cdot}),$$

we obtain the condition that $\boldsymbol{\beta}_{i,\cdot}$ becomes $\mathbf{0}$ if $\|\boldsymbol{s}_i\|_2 \leq \lambda$. Therefore, we can iteratively apply the following equation to update $\boldsymbol{\beta}_{i,\cdot}$ for each $i = 1, \ldots, \tau - 1$:

$$\boldsymbol{\beta}_{i,\cdot} \leftarrow \frac{1}{\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i}}\Big(1 - \frac{\lambda}{\|\boldsymbol{s}_i\|_2}\Big)_+ \boldsymbol{s}_i$$

where $(\cdot)_+ = \max(\cdot, 0)$. The matrix $\boldsymbol{X} \in \mathbb{R}^{\tau\times(\tau-1)}$ is constructed from the position dependent weight $\boldsymbol{d} \in \mathbb{R}^{\tau-1}$.

D Practical Guidelines

As in Boyd et al. (2011), we also update the penalty parameter α to improve convergence and to reduce reliance on its initial choice. After the completion of an ADMM iteration, we calculate the respective primal and dual residuals:

$$r^{(a)}_{\text{primal}} = \sqrt{\frac{1}{\tau \times p}\sum_{i=1}^{\tau}\sum_{j=1}^{p}(\boldsymbol{\theta}^{(a)}_{ij} - \boldsymbol{z}^{(a)}_{ij})^2} \text{ and } r^{(a)}_{\text{dual}} = \sqrt{\frac{1}{\tau \times p}\sum_{i=1}^{\tau}\sum_{j=1}^{p}(\boldsymbol{z}^{(a)}_{ij} - \boldsymbol{z}^{(a-1)}_{ij})^2}$$

at the ath ADMM iteration. We update the penalty parameter α and the scaled dual variable $\boldsymbol{u}$ with the following schedule:

$$\alpha^{(a+1)} = 2\alpha^{(a)}, \boldsymbol{u}^{(a+1)} = \frac{1}{2}\boldsymbol{u}^{(a)} \text{ if } r^{(a)}_{\text{primal}} > 10 \times r^{(a)}_{\text{dual}},$$
$$\alpha^{(a+1)} = \frac{1}{2}\alpha^{(a)}, \boldsymbol{u}^{(a+1)} = 2\boldsymbol{u}^{(a)} \text{ if } r^{(a)}_{\text{dual}} > 10 \times r^{(a)}_{\text{primal}}.$$

Since STERGM is a probability distribution for the dynamic networks, in this work we stop ADMM learning until

$$\left|\frac{l(\boldsymbol{\theta}^{(a+1)}) - l(\boldsymbol{\theta}^{(a)})}{l(\boldsymbol{\theta}^{(a)})}\right| \leq \epsilon_{\text{tol}} \tag{39}$$

where ϵ_{tol} is a tolerance for the stopping criteria.

Algorithm 1 Group Fused Lasso STERGM

Input: initialized parameters $\boldsymbol{\theta}^{(1)}$, $\boldsymbol{\gamma}^{(1)}$, $\boldsymbol{\beta}^{(1)}$, $\boldsymbol{u}^{(1)}$, tuning parameter λ, penalty parameter α, number of iterations for ADMM, Newton-Raphson, and Group Lasso A, C, D, vectorized network data $\vec{\boldsymbol{y}}$, network change statistics $\boldsymbol{H}$
for $a = 1, \cdots, A$ **do**
  $\vec{\boldsymbol{\theta}} = \text{vec}_{\tau p}(\boldsymbol{\theta}^{(a)})$, $\vec{\boldsymbol{z}}^{(a)} = \text{vec}_{\tau p}(\mathbf{1}_{\tau,1}\boldsymbol{\gamma}^{(a)} + X\boldsymbol{\beta}^{(a)})$, $\vec{\boldsymbol{u}}^{(a)} = \text{vec}_{\tau p}(\boldsymbol{u}^{(a)})$
  for $c = 1, \cdots, C$ **do**
    Let $\vec{\boldsymbol{\theta}}_{c+1}$ be updated according to (12)
  end for
  $\boldsymbol{\theta}^{(a+1)} = \text{vec}^{-1}_{\tau,p}(\vec{\boldsymbol{\theta}}_{c+1})$
  Set $\tilde{\boldsymbol{\gamma}} = \boldsymbol{\gamma}^{(a)}$ and $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(a)}$
  for $d = 1, \cdots, D$ **do**
    Let $\tilde{\boldsymbol{\beta}}^{d+1}_{i,\cdot}$ be updated according to (13) for $i = 1, \cdots, \tau - 1$
    $\tilde{\boldsymbol{\gamma}}^{d+1} = (1/\tau)\mathbf{1}_{1,\tau} \cdot (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}^{d+1})$
  end for
  $\boldsymbol{\gamma}^{(a+1)} = \tilde{\boldsymbol{\gamma}}^{d+1}$, $\boldsymbol{\beta}^{(a+1)} = \tilde{\boldsymbol{\beta}}^{d+1}$, $\boldsymbol{z}^{(a+1)} = \mathbf{1}_{\tau,1}\boldsymbol{\gamma}^{(a+1)} + \boldsymbol{X}\boldsymbol{\beta}^{(a+1)}$
  $\boldsymbol{u}^{(a+1)} = \boldsymbol{\theta}^{(a+1)} - \boldsymbol{z}^{(a+1)} + \boldsymbol{u}^{(a)}$
end for
$\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{(a+1)}$
Output: learned parameters $\hat{\boldsymbol{\theta}}$

The algorithm to solve (5) via ADMM is presented in Algorithm 1. The complexity of an iteration for the Newton-Raphson method is $O(\tau^2 p^2)$ and that for the block coordinate descent method is $O(\tau(\tau-1)p)$. In general, the complexity of Algorithm 1 is at least of order $O(A[C\tau^2 p^2 + D\tau(\tau-1)p])$, where A, C, and D are the respective numbers of iterations for ADMM, Newton-Raphson, and Group Lasso. Next, we provide practical guidelines for our proposed method.

By convention, we also implement two post-processing steps to finalize the detected change points $\{\hat{B}_k\}_{k=1}^{K}$. When the spacing between consecutive change points is less than a threshold or $\hat{B}_k - \hat{B}_{k-1} < \delta_{\text{spc}}$, we keep the detected change point with greater $\Delta\hat{\boldsymbol{\zeta}}$ value to avoid clusters of nearby change points. Furthermore, as the endpoints of a time span are usually not of interest, we discard the change point $\hat{B}_k$ less than a threshold

δ_{end} and greater than $T - \delta_{\text{end}}$. In Section 5, we set $\delta_{\text{spc}} = 5$, and we set $\delta_{\text{end}} = 5$ and $\delta_{\text{end}} = 10$ for the simulated and real data experiments, respectively.

E Network Statistics in Experiments

In this section, we provide the formulations of the network statistics used in the simulation and real data experiments. The network statistics of interest are chosen from the extensive list in `ergm` (Handcock et al., 2022), an R library for network analysis. Tables 9 and 10 display the formulations of network statistics used in the respective formation and dissolution models of our method for $t = 2, \ldots, T$. Moreover, Table 11 displays the formulations of network statistics used in the competitor methods for $t = 1, \ldots, T$. The formulations are referred to directed networks, and those for undirected networks are similar.

Table 9: Network statistics used in the formation model

Network Statistics	Formulation of $\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t})$
Edge Count	$\sum_{ij} \boldsymbol{y}_{ij}^{+,t}$
Mutuality	$\sum_{i<j} \boldsymbol{y}_{ij}^{+,t} \boldsymbol{y}_{ji}^{+,t}$
Triangles	$\sum_{ijk} \boldsymbol{y}_{ij}^{+,t} \boldsymbol{y}_{jk}^{+,t} \boldsymbol{y}_{ik}^{+,t} + \sum_{ij<k} \boldsymbol{y}_{ij}^{+,t} \boldsymbol{y}_{jk}^{+,t} \boldsymbol{y}_{ki}^{+,t}$
Homophily	$\sum_{ij} \boldsymbol{y}_{ij}^{+,t} \times \mathbb{1}(\boldsymbol{x}_i = \boldsymbol{x}_j)$
Isolates	$\sum_i \mathbb{1}\big(\deg_{\text{in}}(\boldsymbol{y}^{+,t}, i) = 0 \wedge \deg_{\text{out}}(\boldsymbol{y}^{+,t}, i) = 0\big)$

Table 10: Network statistics used in the dissolution model

Network Statistics	Formulation of $\boldsymbol{g}^{-}(\boldsymbol{y}^{-,t})$
Edge Count	$\sum_{ij} \boldsymbol{y}_{ij}^{-,t}$
Mutuality	$\sum_{i<j} \boldsymbol{y}_{ij}^{-,t} \boldsymbol{y}_{ji}^{-,t}$
Triangles	$\sum_{ijk} \boldsymbol{y}_{ij}^{-,t} \boldsymbol{y}_{jk}^{-,t} \boldsymbol{y}_{ik}^{-,t} + \sum_{ij<k} \boldsymbol{y}_{ij}^{-,t} \boldsymbol{y}_{jk}^{-,t} \boldsymbol{y}_{ki}^{-,t}$
Homophily	$\sum_{ij} \boldsymbol{y}_{ij}^{-,t} \times \mathbb{1}(\boldsymbol{x}_i = \boldsymbol{x}_j)$
Isolates	$\sum_i \mathbb{1}\big(\deg_{\text{in}}(\boldsymbol{y}^{-,t}, i) = 0 \wedge \deg_{\text{out}}(\boldsymbol{y}^{-,t}, i) = 0\big)$

Table 11: Network statistics used in the competitor methods

Network Statistics	Formulation of $\boldsymbol{g}(\boldsymbol{y}^{t})$
Edge Count	$\sum_{ij} \boldsymbol{y}_{ij}^{t}$
Mutuality	$\sum_{i<j} \boldsymbol{y}_{ij}^{t} \boldsymbol{y}_{ji}^{t}$
Triangles	$\sum_{ijk} \boldsymbol{y}_{ij}^{t} \boldsymbol{y}_{jk}^{t} \boldsymbol{y}_{ik}^{t} + \sum_{ij<k} \boldsymbol{y}_{ij}^{t} \boldsymbol{y}_{jk}^{t} \boldsymbol{y}_{ki}^{t}$
Homophily	$\sum_{ij} \boldsymbol{y}_{ij}^{t} \times \mathbb{1}(\boldsymbol{x}_i = \boldsymbol{x}_j)$
Isolates	$\sum_i \mathbb{1}\big(\deg_{\text{in}}(\boldsymbol{y}^{t}, i) = 0 \wedge \deg_{\text{out}}(\boldsymbol{y}^{t}, i) = 0\big)$