Stochastic gradient descent-based inference for dynamic network models with attractors

Hancong Pan, Xiaojing Zhu, Cantay Caliskan, Dino P. Christenson, Konstantinos Spiliopoulos, Dylan Walker & Eric D. Kolaczyk

View supplementary material ⧉

Accepted author version posted online: 06 Jan 2025.

Submit your article to this journal ⧉

Article views: 21

View related articles ⧉

View Crossmark data ⧉

Stochastic gradient descent-based inference for dynamic network models with attractors

Hancong Pan[a], Xiaojing Zhu[a], Cantay Caliskan[b], Dino P. Christenson[c], Konstantinos Spiliopoulos[a], Dylan Walker[d], Eric D. Kolaczyk[e,*]

[a]Department of Mathematics and Statistics, Boston University

[b]Goergen Institute for Data Science, University of Rochester

[c]Department of Political Science, Washington University in St. Louis

[d]Argyros School of Business and Economics, Chapman University

[e]Department of Mathematics and Statistics, McGill University

*E-mail: eric.kolaczyk@mcgill.ca

Abstract

In Coevolving Latent Space Networks with Attractors (CLSNA) models, nodes in a latent space represent social actors, and edges indicate their dynamic interactions. Attractors are added at the latent level to capture the notion of attractive and repulsive forces between nodes, borrowing from dynamical systems theory. However, CLSNA reliance on MCMC estimation makes scaling difficult, and the requirement for nodes to be present throughout the study period limit practical applications. We address these issues by (i) introducing a Stochastic gradient descent (SGD) parameter estimation method, (ii) developing a novel approach for uncertainty quantification using SGD, and (iii) extending the model to allow nodes to join and leave over time. Simulation results show that our extensions result in little loss of accuracy compared to MCMC, but can scale to much larger networks. We apply our approach to the longitudinal social networks of members of US Congress on the social media platform X. Accounting for node dynamics overcomes selection bias in the network and uncovers uniquely and increasingly repulsive forces within the Republican Party. Supplemental materials for the article are available online.

1 Introduction

We consider the problem of modeling dynamic networks, a collection of network graphs G_t indexed over times t. In particular, we focus on a specific class of temporal network models known as CLSNA models, first developed in Zhu et al. (2023). In a CLSNA model, the relations and interactions between nodes and certain attributes of the nodes influence each other as they co-evolve over time.

Latent space network models are a commonly used class of models for static networks (Hoff et al., 2002) where the probability of a relation between actors depends on the positions of individuals in an unobserved "social space" or latent space. Several dynamic latent space network models have been proposed in the literature, which involve embedding nodes of temporal networks into a latent Euclidean space, thus allowing each actor to have a temporal trajectory in the latent space. Earlier approaches modeling the transition and evolution of actors dictate that latent positions evolve over time in a Markov fashion (Sarkar and Moore, 2005; Sewell and Chen, 2015a,b, 2016). In the CLSNA model (Zhu et al., 2023), temporal evolution of latent positions depends on network connectivity via the presence of attractors (a concept that is fundamental to dynamical systems) at the latent level. The CLSNA model has been shown to be effective at disentangling positive (attractive) and negative (repulsive) forces among political elites and the public when applied to longitudinal social networks from the social media platforms X and Reddit (Zhu et al., 2023).

The CLSNA model has adeptly captured the nuances of polarization in American politics, offering valuable insights into the past decade. Despite its efficacy, the model's potential could be further realized by advancing the inference methods to address the challenges of scalability. The challenges of scalability arise in two key aspects. The first aspect is an increasing number of nodes. The second is changes to sets of nodes over time.

The first aspect addresses the reliance on MCMC, as used in Zhu et al. (2023), for estimating model parameters and latent positions. This approach, even though it is quite accurate, becomes prohibitively computationally expensive when scaling to networks with more than a couple hundred nodes. Various methods have been proposed to reduce computational costs for dynamic latent position network models. Sarkar and Moore (2005) introduced a two-stage procedure to optimize the likelihood: multidimensional scaling followed by conjugate gradient update rule. Raftery et al. (2012) used the case-control likelihood approximation to speed up the estimation algorithm. In Liu and Chen (2021), the authors propose to use a variational inference algorithm for estimating the dynamic latent position of the network model.

For the second aspect, the original formulation of CLSNA models assumes that all nodes are present at all time points. In the real world of dynamic networks, the sets of nodes tend to change over time. Sewell and Chen (2015b) develop a model for dynamic networks in which only a subset of edges are observed and the edges are missing at random. Zhu et al. (2023) handles the issue by pre-processing the data and keeping a subset of congress members who are present during the whole period of time studied.

In this paper, we directly address the above two challenges of scalability. Specifically, we (i) develop a Stochastic Gradient Descent (SGD) parameter estimation method that allows us to scale to larger networks; (ii) accompany that by a novel corresponding approach to variance estimation that builds on the notion of Laplace approximation; and (iii) implement these advances within an extension of the CLSNA framework that allows nodes to join and leave the network. Motivated by Laplace's approximation we replace the true posterior probability distribution by a Gaussian, with the mean at the MAP solution and precision matrix determined by the observed Fisher information. Rue et al. (2009) developed the class of INLA algorithms to perform Laplace's approximation when the precision matrix is sparse, for instance in the case of the conditional independence network structure in state-space models and Gaussian Markov Random Fields (GMRFs). However, in the class of CLSNA models that we study here, the assumption of sparsity does not remain applicable, necessitating the novel development we offer here.

We use the Congressional Hashtag Network from the platform formerly known as Twitter, now referred to as "X", as detailed in Zhu et al. (2023). The network was constructed from tweets by US congress-persons from 2010 to 2020. Each year, a binary network was formed, wherein nodes represented sitting members of Congress. Edges between two nodes indicated that their common hashtag usage exceeded the annual average among all pairs of congresspersons. We will revisit the X network to highlight the importance of the two aspects of scalability we propose in this paper: a flexible extended model that accommodates varying sets of nodes, and a fast and scalable SGD-based model inference method. We note that the simplicity and effectiveness of the proposed SGD-based inference and uncertainty quantification developed in this paper is potentially a valuable methodology for generic statistical models. We plan to investigate this direction in future works.

The rest of the paper is organized as follows. Section 2 reviews the standard CLSNA model in Zhu et al. (2023) and presents the extended CLSNA model that enables us to accommodate real world dynamic networks with nodes entering and leaving the networks. Section 3 introduces the two-stage SGD-based inference algorithm: the SGD-based point estimation method and SGD-based variance estimation method. Section 4 provides results from simulations with factorial designs to assess the capability of the proposed algorithm under different settings. [1] In Section 5, we report the results from fitting the extended model to the X Congressional Hashtag Network.

2 Extended CLSNA model

The CLSNA model of Zhu et al. (2023) assumes that all nodes are present at all time points. In the real world of dynamic networks, the sets of nodes tend to change over time. Take the X Congressional Hashtag Network as an example. Figure 1 shows numbers of re-elected and newly elected Democratic and Republican congress members in the X network. We observe significant number of changes in the sets of nodes present at each time because congress members only serve a fixed-year term before they are considered for reelection and because newly elected or sitting congress members decided to open an X account.

In order to use the original CLSNA model, Zhu et al. (2023) keep a subset of congress members who were in office and on X during the whole period of time. After applying this filtering criteria, the number of nodes was reduced from around 500 to 207.

By developing a model that accommodates nodes entering and leaving the network, we could make inference based on the full data set. As a result, we are able to obtain latent position estimates at each time and estimates of the attractive or repulsive forces between a given node and its neighbors (influencing its movement over time) that are unbiased by factors like time in office and early technology adoption.

2.1 Model Definition

Let $G_t = (V_t, E_t)$ be a network evolving in (discrete) time t, with vertex set V_t and edge set E_t. We allow for V_t to vary over time. Let Y_t be the (random) adjacency matrix at time t corresponding to G_t. Data is in the form of time series of adjacency matrices $\{y_t : t = 1, \cdots, T\}$, where $y_{t,ij} = 1$ if there is an edge between node i and node j at time t and $y_{t,ij} = 0$ otherwise.

We model Y_t using a latent space approach, with attractors added in the latent level to capture the notions of attractive and repulsive forces. In addition, we want to accommodate nodes entering and leaving the network. Let $Z_{t,i} \in \mathbb{R}^p$ be a time-indexed latent (i.e., unobserved) position for node i in p-dimensional Euclidean space, and let $Z_t = \{Z_{t,i}\}$. Assume that each of the N_t nodes falls into one of two groups, i.e., Democratic and Republican, with known node label $\pi(i) \in \mathcal{C}$ for node i, where $\mathcal{C} = \{1, 2\}$ is the set of group labels.

Formally, we define our model as follows:

$$Y_{t,ij} \mid p_{t,ij} \sim Bernoulli(p_{t,ij}) \ (1)$$

where at time $t = 1$,

$$\text{logit}(p_{t,ij}) = \alpha - s(z_{t,i}, z_{t,j}) \ (2)$$

$$Z_{t,i} \sim Normal(0, \tau^2 I_p) \ . \ (3)$$

If, at time $t \geq 2$, if node i was absent from the network at time $t-1$,

$$\text{logit}(p_{t,ij}) = \alpha + \delta Y_{t-1,ij} - s(z_{t,i}, z_{t,j}) \ (4)$$

$$Z_{t,i} \sim Normal(\beta_{t,i} Z_{t_0,i} + (1 - \beta_{t,i})\mu_{t,i}, \phi^2 I_p) \ (5)$$

$$\mu_{t,i} = \overline{z}_{t-1}^{\pi(i)} \ , \ (6)$$

where $\beta_{t,i} \in [0,1]$ and $Z_{t_0,i}$ denotes the potential position of the specific node i at the most recent past time $t_0 < t-1$ that it was present at the network. If node i has never been in the network before time t, then we simply set $\beta_{t,i} = 0$. See also Remark 2.1.

If at time $t \geq 2$, node i was present at the network at time $t-1$,

$$\text{logit}(p_{t,ij}) = \alpha - s(z_{t,i}, z_{t,j}) \quad (7)$$

$$Z_{t,i} \mid Z_{t-1,i} = z_{t-1,i} \sim Normal(\mu_{t,i}, \sigma^2 I_p) \quad (8)$$

$$\mu_{t,i} = z_{t-1,i} + \gamma_w^{\pi(i)} A_i^w(z_{t-1}, Y_{t-1}) + \gamma_b A_i^b(z_{t-1}, Y_{t-1}) \quad . \quad (9)$$

Here $s(\cdot, \cdot)$ is the Euclidean distance, $s(z_{t,i}, z_{t,j}) = \| z_{t,i} - z_{t,j} \|_2$. A_i^w and A_i^b are the within group $\pi(i)$ (the group node i belongs to) and between groups attractor functions for node i in Y_{t-1} respectively. In addition, $\gamma_w^{\pi(i)}$ and γ_b are the attraction parameters within group $\pi(i)$ and between groups attraction parameters respectively. The two attractors for node i are defined as follows [2],

$$A_i^w(z_{t-1}, Y_{t-1}) = \overline{z}_{t-1,i}^1 - z_{t-1,i}, \overline{z}_{t-1,i}^1 = \frac{1}{\mid S_1^{(i)} \mid} \sum_{j \in S_1^{(i)}} z_{t-1,j} \quad (10)$$

$$A_i^b(z_{t-1}, Y_{t-1}) = \overline{z}_{t-1,i}^2 - z_{t-1,i}, \overline{z}_{t-1,i}^2 = \frac{1}{\mid S_2^{(i)} \mid} \sum_{j \in S_2^{(i)}} z_{t-1,j} \quad , \quad (11)$$

which are the discrepancies of $z_{t-1,i}$ from two local averages at time $t-1$. The sets $S_1^{(i)}$ and $S_2^{(i)}$ are defined based on group membership and network connectivity as follows:

1. $S_1^{(i)} = \{ j \in \mathcal{N} \setminus i \mid Y_{ij} = 1, \pi(i) = \pi(j) \}$, neighbors of node i in the same group

2. $S_2^{(i)} = \{ j \in \mathcal{N} \setminus i \mid Y_{ij} = 1, \pi(i) \neq \pi(j) \}$, neighbors of node i in a different group.

Note that even though it is not explicitly mentioned in the notation, the sets $S_1^{(i)}$ and $S_2^{(i)}$ also depend on time because the connectivity sets are allowed to change with time. When $S_1^{(i)}$ or $S_2^{(i)}$ or both are empty, we set $A_i^w(z_{t-1}, Y_{t-1}) = 0$, or $A_i^b(z_{t-1}, Y_{t-1}) = 0$ or both to be zero respectively. This implies that if node i is not connected to any members of a specific group, it receives no force from that group.

Remark 2.1 .

We remark that (5) allows to cover the case where a given node i leaves the network at a given time t_0 and re-enters again at a future time $t > t_0 + 1$. In that case, (5) shows that the mean of a new position can be defined to be a convex combination of the last known position and the average of the position of rest of the nodes within the group. If no further information exists, taking $\beta_{t,i} = \frac{1}{2}$ is a reasonable choice. Of course, if node i enters the network for the first time at time t, then the intuitive choice is to take $\beta_{t,i} = 0$.

Note that we automatically recover the original CLSNA model when $V_t \equiv V$ is fixed over time. In this case, at time $t \geq 2$, a node i is always present at time $t-1$ and (4) and (5) are redundant in that case.

In this proposed extended model, each node lies in a p-dimensional Euclidean latent space. The smaller the distance between two nodes in the latent space, the greater their probability of being connected, as in (1), (2), (4), and (7). The expressions in Eqs. (10) and (11) capture the discrepancy between the current latent position of node i and the average of that of its current neighbors in groups 1 and 2, respectively. The corresponding parameters γ_1^w, γ_2^w, and γ^b represent attractive and repulsive forces respectively. CLSNA model allows the network connectivity to enter the temporal evolution of latent positions in the form of attractors. Specifically, in the proposed model the evolution of latent positions for each node i from $t-1$ to t is modeled by the normal transition distribution in Eq. (9), the mean vector of which depends not only on the latent position of itself at time $t-1$, but also on the two local averages, one from its neighbors in the same group, the other from its neighbors in a different group, as captured in (10) and (11). Strength of attraction and repulsion toward local averages is summarized by the attractor functions and the associated parameters. The parameter δ captures edge persistence. For $\delta > 0$, the probability of an edge at time t will be increased when one exists already at time $t-1$.

The extended model is a natural extension of the original model. If there are no nodes entering or leaving the networks, the extended model is identical to the original model of Zhu et al. (2023). When the sets of nodes change over time, the model classifies nodes into two categories and models the two types of nodes separately:

1. For actors who are "retained" from time $t-1$ to t, the manner in which network connectivity influences the temporal evolution of latent positions remains consistent with the original model: the evolution of latent positions for the node i from $t-1$ to t is modeled by the normal transition distribution, the mean vector of which is a linear combination of the latent position of itself at time $t-1$ and the mean of both its neighbors in the same group and its neighbors in a different group.

2. For an actor that is present at time t but is absent at time $t-1$, we cannot make an educated guess of its current position at time t because of its absence at time $t-1$. For this reason we choose to model the prior distribution of its latent position as a normal distribution, the mean vector of which is the mean of all members in the same group at $t-1$, which we denote by $\bar{z}_{t-1}^{\pi(i)}$.

We also present a graphical representation of the proposed model in Figure 2. The graphical representation is adapted from Zhu et al. (2023). This diagram illustrates the key dependencies in the model. The arrow from $Y_{t-1,ij}$ to $Y_{t,ij}$ represents the edge persistence effect described by Eq. (4), indicating that the presence of an edge at time $t-1$ influences the likelihood of an edge at time t. The arrow from $Y_{t-1,ij}$ to Z_t reflects the influence of prior network connectivity on the evolution of latent positions, as captured by Eq. (8). The generation of edges based on latent position similarity, as described in Eq. (1), is depicted by the arrows from Z_t to $Y_{t,ij}$. Finally, the arrow from Z_{t-1} to Z_t illustrates the influence of the previous position, i.e., at time $t-1$ on the position at time t.

3 SGD-based Inference Method

We develop a two-stage algorithm to make inferences based on the posterior $\pi(\theta, Z_{1:T} \mid Y_{1:T})$, where $\theta = (\alpha, \delta, \gamma^w, \gamma^b)$. At the first stage we use SGD to compute a point estimate. At the second stage, we estimate marginal posterior standard deviations by a novel approach to quadratic approximation of the log-posterior density, which is also based on SGD.

3.1 Point Estimate

The posterior distribution of latent positions and parameters is

$$
\begin{aligned}
&\pi(Z_{1:T}, \theta | Y_{1:T}) \\
&\propto P(Y_{1:T}, Z_{1:T} | \theta)\pi(\theta) \\
&\propto \left(\prod_{t=2}^{T} P(Z_t | Z_{t-1}, Y_{t-1}) P(Y_t | Z_t, Y_{t-1}) \right) P(Y_1 | Z_1) P(Z_1)\pi(\theta) \\
&= \left(\prod_{t=2}^{T} \prod_{i=1}^{N_t} \left(P(Z_{t,i} | Z_{t-1}, Y_{t-1}) \prod_{j:j\neq i} P(Y_{t,ij} | Z_{t,i}, Z_{t,j}, Y_{t-1,ij}) \right) \right) \cdot \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad P(Y_1 | Z_1) P(Z_1)\pi(\theta).
\end{aligned}
\tag{12}
$$

The log posterior distribution (i.e., taking the log of (12)), therefore, can be written as sums of simpler functions. To optimize the posterior function with SGD, at each step, we randomly sample different terms from these sums. The key steps are:

1. Initialize $Z_{1:T}, \theta$.

2. Randomly sample indices of different terms from the different summands of the log-posterior distribution function. Compute the randomly sampled summand functions of the log-posterior distribution function.

3. Update $Z_{1:T}, \theta$ based on stochastic gradient descent using the output from step 2.

Repeat steps 2-3 until convergence criterion is met. In practice, we use gradient descent when the sample size is small and it is computationally feasible. In our implementation, we used gradient descent with momentum (when the sample size was small) or stochastic gradient descent with momentum (when the sample size was large), see Polyak (1964).

Algorithm 1 SGDSTEP

Input: Network time-series $Y_{1:T}$. Current values $\theta^{(k)}$, $Z_{1:T}^{(k)}$ for parameters and latent positions.

λ : step size for $Z_{1:T}$ and for θ

Output: $\theta^{(k+1)}, Z_{1:T}^{(k+1)}$

1: Set θ, $Z_{1:T}$ to be $\theta^{(k)}$, $Z_{1:T}^{(k)}$.

2: Uniformly sample the terms (factors) from the posterior distribution:

$$\prod_{t=2}^{T}\prod_{i=1}^{N} P(Z_{t,i}|Z_{t-1},Y_{t-1})\prod_{j:j\neq i} P(Y_{t,ij}|Z_{t,i},Z_{t,j},Y_{t-1,ij})P(Y_1|Z_1)P(Z_1)\pi(\theta) \quad (13)$$

Denote the sampled terms by p_i. Define $\pi_{SGD}(\theta, Z_{1:T} | Y_{1:T})$ to be $\prod_i p_i$

3: Take a SGD step (see also Remark 3.2)

$$Z_{1:T}^{(k+1)} = Z_{1:T}^{(k)} - \lambda\left(\frac{\partial \log \pi_{SGD}(\theta^{(k)}, Z_{1:T}^{(k)} | Y_{1:T})}{\partial Z_{1:T}}\right)$$

$$\theta^{(k+1)} = \theta^{(k)} - \lambda\left(\frac{\partial \log \pi_{SGD}(\theta^{(k)}, Z_{1:T}^{(k)} | Y_{1:T})}{\partial \theta}\right) \quad (14)$$

Formally, Algorithm 1 is the main building block for the SGD-based point estimation method, which we present as pseudo-code in Algorithm 2.

Algorithm 2 SGD-BASED POINT ESTIMATION METHOD

Input: Network time-series $Y_{1:T}$. Initial values θ^{init}, $Z_{1:T}^{init}$ for parameters and latent positions. M: maximum number of iterations, λ: step size for $Z_{1:T}$ and for θ

Output: $\theta^{(k)}, Z_{1:T}^{(k)}$

1: Set initial values $\theta^{(0)}$, $Z_{1:T}^{(0)}$ to be θ^{init}, $Z_{1:T}^{init}$.

2: **for** $k \leftarrow 0$ to *M* **do**

3: Take a SGD step to optimize the posterior distribution: $(\theta^{(k+1)}, Z_{1:T}^{(k+1)})$ = SGDSTEP($\theta^{(k)}, Z_{1:T}^{(k)}$)

4: Stop if stopping criteria is met

5: **end for**

Remark 3.1 .

In our work using gradient descent, we encountered a recurring challenge. For the majority of time points, the algorithm performs as expected, closely matching the true underlying values in simulation. However, there were a couple of time points at which an anomalous sign inversion occurs in one dimension of the latent position. In particular, if $z_{t,i} = [z_{t,i}^1, z_{t,i}^2, \ldots, z_{t,i}^p]$

was the true value, it was estimated to be $\tilde{z}_{t,i} = [-z_{t,i}^1, z_{t,i}^2, \ldots, z_{t,i}^p]$, even though all the estimates for other times match the truth.

This problem is rooted in the complex landscape of our problem, where the model sometimes settles in local modes due to its non-convex nature. The issue with local modes is similar to the issue of label-switching found in the dynamic stochastic block model literature and the issue may not be solved without an extra assumption e.g. that most of the nodes do not change group across two different time steps (Matias and Miele, 2017).

The resolution to this issue that we found to work well is to start with a CLSNA model with latent variables living in a higher dimensional space from the one we intend on having; that is, we aim to fit a model with $Z \in \mathbb{R}^p$ but we intentionally choose $Z \in \mathbb{R}^q$ with $q > p$. Then, we use the first p principal components of the vector of the latent position variables for $Z_{1:T}$. This method was the most effective of several alternatives attempted to remedy local modes. Intuitively, one can think of this approach as creating a path between modes thus allowing SGD to travel between modes in the higher dimension. In both the simulated and real data examples, we initially set p to 2 and chose q as 3. We begin by fitting a model with Z in $\mathbb{R}^3$, followed by applying principal component analysis (PCA) to the resulting latent positions, thereby compressing Z into $\mathbb{R}^2$ while maximizing information preservation. This reduced form is then used as the initial value for fitting a model with Z in $\mathbb{R}^2$. This approach, albeit doubling the cost of fitting the model and performing PCA, provides more accurate and faster convergence towards the true values, as it effectively navigates through potential local optima that could hinder convergence in a single-stage fitting process.

Remark 3.2 .

Motivated by the work in Hinton (2012), at the initial stage of Algorithm 1 in Eq.(14) we use the sign of the gradient of the global parameters (+1/-1), i.e., we replace (14) by

$$\theta^{(k+1)} = \theta^{(k)} - \lambda \ \mathrm{sign}\left(\frac{\partial \log \pi_{SGD}(\theta^{(k)}, Z_{1:T}^{(k)} | Y_{1:T})}{\partial \theta} \right).$$

This is due to the fact that the magnitude and variance of the related gradient terms is much larger than the magnitude and variance of the gradients of the latent positions. This leads to more stability. As we approach the conclusion of the training phase, we switch to Eq.(14) for better accuracy.

3.2 Variance Estimation

The first stage with SGD above only gives point estimates, without quantifying uncertainty. We propose a novel approach to estimate the posterior variance for parameters of interest. It is a method for approximate Bayesian inference based on Laplace's method. Laplace's approximation, when applied in scenarios like state-space models and Gaussian Markov Random Fields, leverages the sparsity of the precision matrix as detailed by Rue et al. (2009). However, in more general contexts, such as CLSNA, this assumption of sparsity is not always valid. In response to this, our proposed approach builds upon Laplace's foundational principles but adapts to the challenges of non-sparse structures. Specifically, it transforms the variance estimation problem into an optimization problem that can also be solved with SGD.

In this section, we propose a novel variance estimation algorithm. Our variance estimation method exploits the properties of the conditional distribution of a multivariate Gaussian distribution. Let x follow a multivariate normal distribution $x \sim \mathcal{N}(\mu, \Sigma)$, where the block-wise mean and covariance matrices are defined as $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$. The conditional distribution of a subset vector x_1, given its complement vector x_2, is also a multivariate normal distribution $x_1 \mid x_2 \sim \mathcal{N}(\mu_{1|2}, \Sigma_{1|2})$. We will use $\mu_{1|2}(x_2)$ to denote the conditional mean of x_1 given x_2. With this notation, the variance and covariance can be respectively calculated as:

$$\mathrm{Var}(x_2) = \frac{1}{2} \frac{(\mu_2 - \tilde{x}_2)^2}{\log p(x_1 = \mu_1, x_2 = \mu_2) - \log p(x_1 = \mu_{1|2}(\tilde{x}_2), x_2 = \tilde{x}_2)}, \quad (15)$$

$$\Sigma_{12} = \frac{\Sigma_{22}}{(\tilde{x}_2 - \mu_2)} (\mu_{1|2}(\tilde{x}_2) - \mu_1). \quad (16)$$

A more detailed discussion and proof of these properties can be found in the supplementary file. The results above for a generic normal distribution $\mathcal{N}(\mu, \Sigma)$ are now adapted to estimate variance in the posterior distribution $\pi(\cdot \mid Y)$, effectively applying the same theoretical concepts to the Bayesian analysis under a Laplace approximation.

3.2.1 The Variance Estimation Algorithm

Eq.15 and Eq.16 are the primary components in the second stage of our proposed SGD-based method for variance estimation of a single parameter, as detailed in Algorithm 3. The first stage detailed in section 3.1 computes a point estimate using SGD. In the second stage, with this point estimate, we redo the optimization with the constraint of fixing the parameter to be variance-estimated, adjusting it by a certain η perturbation size. Thanks to Eq.15 and Eq.16, we can then estimate the variance/covariance of this parameter by observing the changes in the posterior function and the changes in the other parameters resulting from the perturbation.

Algorithm 3 details the second stage of the SGD-based approach for estimating the variance of a single parameter (in here we choose α).

Algorithm 3 SGD-ʙᴀsᴇᴅ Vᴀʀ/Cᴏᴠ Esᴛɪᴍᴀᴛɪᴏɴ Mᴇᴛʜᴏᴅ ғᴏʀ ᴀ Sɪɴɢʟᴇ Pᴀʀᴀᴍᴇᴛᴇʀ

Input: Network time-series $Y_{1:T}$; mode of $\log \pi(Z, \theta \mid Y)$, θ^* (for all hyperparameters $\alpha, \delta, \gamma^w, \gamma^b$), $Z_{1:T}^*$ (latent positions); M: max iterations; λ: step size for $Z_{1:T}$ and for θ; η_α: perturbation size for α.

Output: $\mathrm{Var}(\alpha \mid Y)$, $\mathrm{Cov}(\alpha, \theta_i \mid Y)$, $\mathrm{Cov}(\alpha, Z_{1:T} \mid Y)$

1: Set initial values $\theta^{(0)}$, $Z_{1:T}^{(0)}$ to θ^*, $Z_{1:T}^*$.

2: **for** $k \leftarrow 0$ to M **do**

3: Set $\alpha^{(k)} = \alpha^* + \eta_\alpha$.

4: $(\theta^{(k+1)}, Z_{1:T}^{(k+1)}) = \text{SGDst ep}(\theta^{(k)}, Z_{1:T}^{(k)})$.

5: Stop if criteria are met; let k be the corresponding iteration.

6: **end for**

7: $\text{Var}(\alpha \mid Y) = \dfrac{1}{2} \dfrac{\eta_\alpha^2}{\log \pi(\theta^{(0)}, Z_{1:T}^{(0)} \mid Y_{1:T}) - \log \pi(\theta^{(k+1)}, Z_{1:T}^{(k+1)} \mid Y_{1:T})}$

8: $\text{Cov}(\alpha, \theta_i \mid Y) = \dfrac{\text{Var}(\alpha \mid Y)}{\eta_\alpha}(\theta_i^{(k+1)} - \theta_i^{(0)})$

9: $\text{Cov}(\alpha, Z_{1:T} \mid Y) = \dfrac{\text{Var}(\alpha \mid Y)}{\eta_\alpha}(Z_{1:T}^{(k+1)} - Z_{1:T}^{(0)})$

In Figure 3 we aim to visualize this approach for a a Bivariate Distribution.

To extend this estimation to all parameters of interest, we repeatedly apply Algorithm 3 individually to each parameter. Since we are able to recover the covariance matrix, we are also able to estimate variance of functions of parameters. As an illustration, let us look for example into $\text{Var}(\gamma^w - \gamma^b)$. This estimation is achieved through a sequential application of Algorithm 3 to each component parameter. Initially, the algorithm is applied to γ^w, yielding estimates $\text{Var}(\gamma^w \mid Y)$ and $\text{Cov}(\gamma^w, \gamma^b \mid Y)$. Following this, a similar application to γ^b provides $\text{Var}(\gamma^b \mid Y)$ and an alternative estimate of $\text{Cov}(\gamma^w, \gamma^b \mid Y)$. The variance of the composite parameter $\gamma^w - \gamma^b$ is then approximated using the formula $\text{Var}(\gamma^w \mid Y) + \text{Var}(\gamma^b \mid Y) - 2\text{Cov}(\gamma^w, \gamma^b \mid Y)$. Given that the function is not exactly quadratic, discrepancies between the two covariance estimates may arise. To mitigate this, one can proceed with averaging these estimates for a more robust estimation.

4 Simulation studies

We simulate two networks, one with $n = 100$ actors and one with $n = 1000$ actors. For each one of them, we simulate two parameter settings, one for positive attraction or 'flocking' ($\gamma^b \geq 0$) and the other for repulsion ($\gamma^b \leq 0$) or 'polarization' between two groups over time. For each setting we simulate 20 data sets with the number of time points $T = 10$.

We want to compare the mean of the point estimates over 20 simulations with the truth and we compare the mean of the estimated variances with the standard deviation of the point estimates over 20 simulations.

With the number of nodes $n = 100$, Table 1 shows that the point estimates are reasonably accurate compared with the truth. In the flocking setting, all estimates are slightly biased to

be smaller than the truth. In the polarization setting, all estimates except for repulsion between groups (γ^b) are slightly biased to be smaller than the truth. The variance estimates are reasonably accurate and are generally slightly smaller than the true standard deviation (See Table 1). The empirical standard deviations are obtained by 20 MCMC trials in each parameter setting. (See Table 1, where the empirical standard deviations are shown in parenthesis).

Table 1 shows that when the number of nodes is $n = 1000$, the point estimates are more accurate than when the number of nodes $n = 100$, both in the flocking setting and the polarization setting, and similarly for the variance estimates.

We notice that the estimation of the standard deviation slightly underestimates the true standard deviation. One reason is that the posterior function can have more than one local minimum. This class of variation estimation methods is able to approximate the contribution of one local minimum to the total variance. Note that multiple local minima contribute to total variance, but in our experiments this did not result into serious issues.

As a comparison of the computational efficiency of the proposed SGD algorithm with MCMC algorithm, in the X data analysis in the following section, on the reduced data set with 200 nodes, the MCMC took about four hours (CPU) while the proposed SGD-based algorithm only took fewer than 10 minutes to run (GPU). The supplementary file includes a more thorough comparison among MCMC and SGD-based estimation (CPU and GPU).

In the supplementary file, we also include a simulation study to demonstrate the algorithm's robustness in the scenario where nodes are allowed to leave the network over time, i.e., in the case where the vertex set V_t changes dynamically in time.

5 X Data Analysis

5.1 Exploratory Analysis

The X Congressional Hashtag Networks dataset in Zhu et al. (2023) was based on 843,907 tweets from 796 US Congresspersons' accounts from 2010-2020. Yearly binary networks were built from the tweets. Nodes represent members and edges show whether two members used common hashtags more than the year's average. The nodes in the yearly networks vary as Congress members' X participation changes due to reelections, late joins, or early departures see Figure 1. Figure 4 shows temporal evolution of edge density within the Democratic and Republican parties individually, alongside the inter-party edge density, as well as the edge density across the entire network. Similar trends were observed in Zhu et al. (2023): connections increased for the first four years for all types of edges. From 2015 to 2020, edge density within the Democratic party continues to rise, while inter-party edge density decays slowly and edge density within the Republican party drops sharply.

Figure 1 shows numbers of re-elected and newly elected Democratic congressman and Republican congressman in the X network. We can see an upward trend in the number of X users in both parties. The number of Democratic X users grows more steadily with fewer fluctuation compared with the Republican X users. Notably, Republicans outnumber Democrats in the X network for 9 out of 11 years.

5.2 Results

We fit our model with time-invariant parameters to the full X data set from year 2010 to 2020. The AUC values computed at each year are all above 0.974, and the overall AUC value computed across all times is 0.985, providing evidence that our model fits the data very well. The point estimates and the posterior standard deviation of model parameters are provided in Table 2. Point estimates were produced using the SGD methodology of Section 3.1 and standard deviation estimates were produced using the methodology of Section 3.2. In the supplementary file, we present two diagnostic tests demonstrating that the underlying assumptions of the variance estimation method proposed in Section 3.2 are largely accurate for this dataset, i.e., the diagnostic tests do not show evidence of violating the underlying assumptions of the variance estimation method.

The point estimates of persistence parameter is 1.08, implying that there is a higher likelihood for it to also appear at time t if an edge is present at time $t-1$. The between-group attraction coefficient γ^b is -0.22, indicating that there is polarization between the two parties. The within group coefficients γ^w are 0.34 for Democrats, and -0.11 for Republicans, indicating that the Democrats were flocking, while the Republicans were polarizing within their own group. Importantly, we note that in the original model in Zhu et al. (2023), only members that are present at all times are kept. The parameter estimation for attractions within group indicated that while both parties have moved away from one another, they generally flocked to their own. By incorporating all members into the extended model, we effectively eliminate the potential for selection bias, ensuring a more accurate and comprehensive analysis of the data. As a result of this approach, we have uncovered that the force within the Republican group is negative, in contrast to Zhu et al. (2023).

Figure 5 shows the point estimates of latent positions for each member of Congress in the X hashtag networks. At each time, two parties consistently occupy different halves of the space. The Democrats were flocking during the whole time. Conversely, the Republicans initially displayed a similar flocking behavior, but gradually began to disperse after year 2017.

Figure 6 shows the trajectory of the mean of the latent positions of the members in each party. The lighter color represents results obtained from the reduced data set used in Zhu et al. (2023)'s original analysis. While the darker color represents results obtained with our model and inference techniques from the full data set. In both cases, the average position of the two parties gradually and consistently approached each other from 2011 to 2016. But after 2016, both of them took a U-turn and moved apart. However, we see that the analysis based on the full data set suggests that the Democrats were in fact noticeably closer to the Republicans just before 2016 than suggested by the analysis based on the reduced data.

Figure 7 compares attraction parameters with that found in Zhu et al. (2023). The MCMC and the SGD-based method yield close point estimates and variance estimates for the attraction parameters on the reduced data set. The point estimates obtained from the MCMC method fall within the 95% confidence interval of the SGD-based method. On the full data set, the model concludes that there is a large attractive force within the Democratic Party. By contrast, we find that within the Republican Party the force is repulsive. To analyze the changes in attraction, repulsion, we fit a series of models that allow a change-point to vary from 2011 to 2019. The resulting fitted models with different change-points were compared based on Bayesian Information Criterion (BIC). The one with the lowest BIC value was

selected. We identified 2012 as the year of change-point (See supplementary file for more details on this). Figure 8 illustrates the evolution of within-group attraction and repulsion for Democrats and Republicans, and between-group attraction and repulsion. The within-Republican coefficient is positive in the first time period and is negative in the second time periods from 2012 to 2020. This suggests polarization within Republican members started to rise within the period 2012-2020. The within-Democrat coefficient is positive in both time periods, although its magnitude decreases a bit in the second time period from 2012 to 2020. The between-group coefficient (orange bars) is not statistically significant at the 95% confidence level from 2010 to 2012 and is negative in the second time periods from 2012 to 2020.

The observed patterns of increasing polarization between the Democratic and Republican parties align with findings from previous studies. For instance, Andris et al. (2015) documented similar trends through the analysis of co-sponsorship patterns in Congress, while Garimella and Weber (2017) identified heightened polarization in the retweet networks of U.S. politicians on Twitter. These studies, along with our findings, highlight political polarization in different environments in U.S..

6 Discussion

The extended CLSNA model proposed in this paper together with the SGD-based inference method, successfully addressed the challenge of scalability to large networks with nodes entering and leaving the network. In particular, (a) the introduction of the SGD parameter estimation method, (b) the development of the novel variance estimation approach for approximate Bayesian inference which transforms variance estimation into an optimization problem solvable with SGD, and, (c) the extension of the model to allow dynamic node participation, have significantly improved the CLSNA model's utility for analyzing dynamic networks. When applied to longitudinal social networks on X, the model mitigates selection bias and reveals previously concealed negative force within the Republican party.

The SGD parameter estimation method and the variance estimation method are general approaches to approximate Bayesian inference. Therefore, they can be combined with other strategies to make inference faster. For example, they can be used with case-control log-likelihood approximation as in Raftery et al. (2012). This can be achieved by using stratified sampling to draw the edge probability terms during the stochastic sampling process.

With the increased efficiency of the proposed method, extending the model to accommodate more than two groups with attractors becomes more feasible. This extension is particularly useful in contexts such as multi-party political systems in Germany and Canada. For a small number of groups, the computational impact is expected to be minimal. Moreover, our library has been developed with this flexibility in mind and allows for such an extension.

In future work we plan to incorporate observed explanatory variable terms into the edge probability. For instance, a linear predictor based on explanatory variables can be integrated into the edge probability function, see Hoff (2007). In this way, we can more effectively separate the latent effect from the fixed effect and accurately capture the repulsive and attractive force among the different actors.

Acknowledgement

Supplementary Materials

Appendix: Contains proofs of the theoretical results, additional statistical tests and diagnostics to support the numerical results, implementation details of the algorithm, performance/efficiency studies of the algorithm, and a simulation study demonstrating the algorithm's robustness against node turnover. (supp.pdf, PDF)

Source Code: The source code is available in the GitHub repository. The current library implementation is specifically designed for a 2-party system.

(https://github.com/KolaczykResearch/SGD4CLSNA, GitHub repository)

Notes

[1] Code for this paper can be found at https://github.com/KolaczykResearch/SGD4CLSNA

[2] This application employs a two-group CLSNA model with attractors that mimic attractive and repulsive force. It is a specific version of the general CLSNA model class. Each node is assigned to one of two labeled groups, and movements are dictated by attractors shaped by neighboring nodes' influence.

References

Andris, C., Lee, D., Hamilton, M. J., Martino, M., Gunning, C. E. and Selden, J. A. (2015) The rise of partisanship and super-cooperators in the us house of representatives. *PloS one*, **10**, e0123507.

Garimella, V. R. K. and Weber, I. (2017) A long-term analysis of polarization on twitter. In *Proceedings of the International AAAI Conference on Web and social media*, vol. 11, 528–531.

Hinton, G. (2012) Neural networks for machine learning. Coursera. URL: `https://www.coursera.org/course/neuralnets`.

Hoff, P. (2007) Modeling homophily and stochastic equivalence in symmetric relational data. *Advances in Neural Information Processing Systems*, **20**.

Hoff, P. D., Raftery, A. E. and Handcock, M. S. (2002) Latent space approaches to social network analysis. *Journal of the American Statistical Association*, **97**, 1090–1098.

Liu, Y. and Chen, Y. (2021) Variational inference for latent space models for dynamic networks. *arXiv preprint arXiv:2105.14093*.

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

Polyak, B. T. (1964) Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, **4**, 1–17.

Raftery, A. E., Niu, X., Hoff, P. D. and Yeung, K. Y. (2012) Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of Computational and Graphical Statistics*, **21**, 901–919.

Rue, H., Martino, S. and Chopin, N. (2009) Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society: Series B (statistical methodology)*, **71**, 319–392.

Sarkar, P. and Moore, A. W. (2005) Dynamic social network analysis using latent space models. *SIGKDD Explorations*, **7**, 31–40.

Sewell, D. K. and Chen, Y. (2015a) Analysis of the formation of the structure of social networks by using latent space models for ranked dynamic networks. *Journal of the Royal Statistical Society: Series C: Applied Statistics*, 611–633.

— (2015b) Latent space models for dynamic networks. *Journal of the American Statistical Association*, **110**, 1646–1657.

— (2016) Latent space models for dynamic networks with weighted edges. *Social Networks*, **44**, 105–116.

Zhu, X., Caliskan, C., Christenson, D. P., Spiliopoulos, K., Walker, D. and Kolaczyk, E. D. (2023) Disentangling positive and negative partisanship in social media interactions using a coevolving latent space network with attractors model. *Journal of the Royal Statistical Society Series A: Statistics in Society*. URL: `https://doi.org/10.1093/jrsssa/qnad008`. Qnad008.
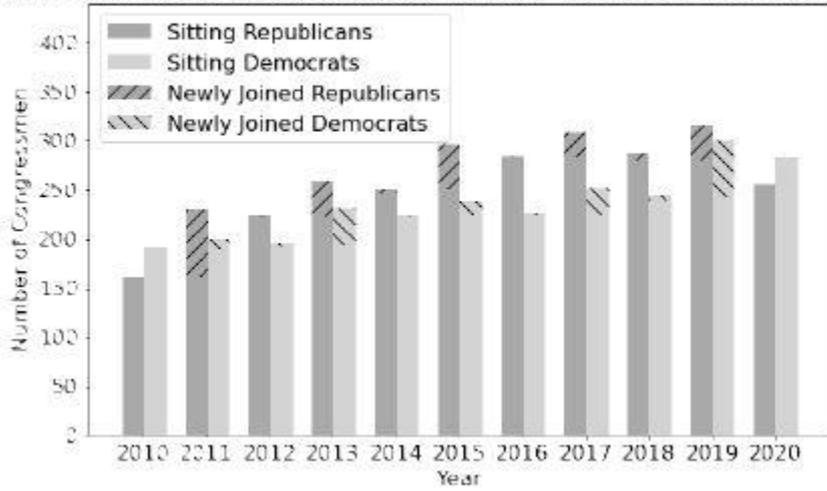
Figure 1: Numbers of re-elected and newly elected Democratic and Republican congress members in the X hashtag network
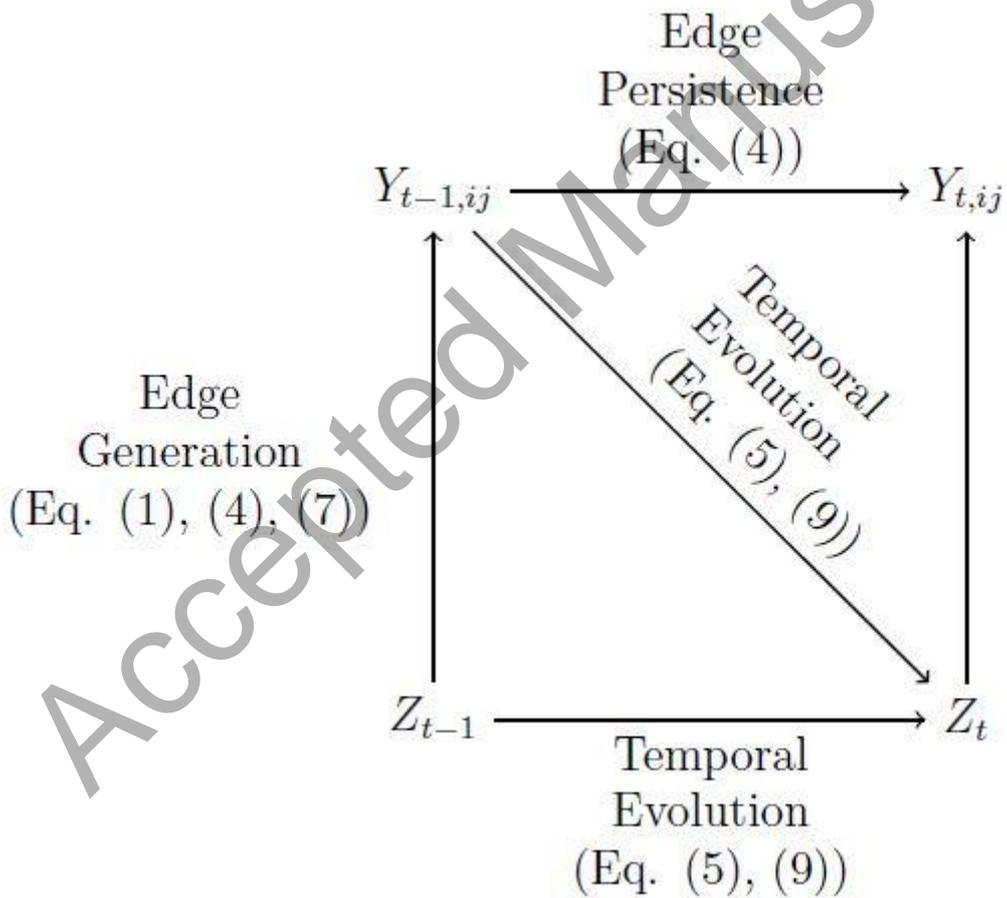

Figure 2: Graphical model representation of the CLSNA model. All key dependencies are shown, including edge generation, edge persistence and temporal evolution
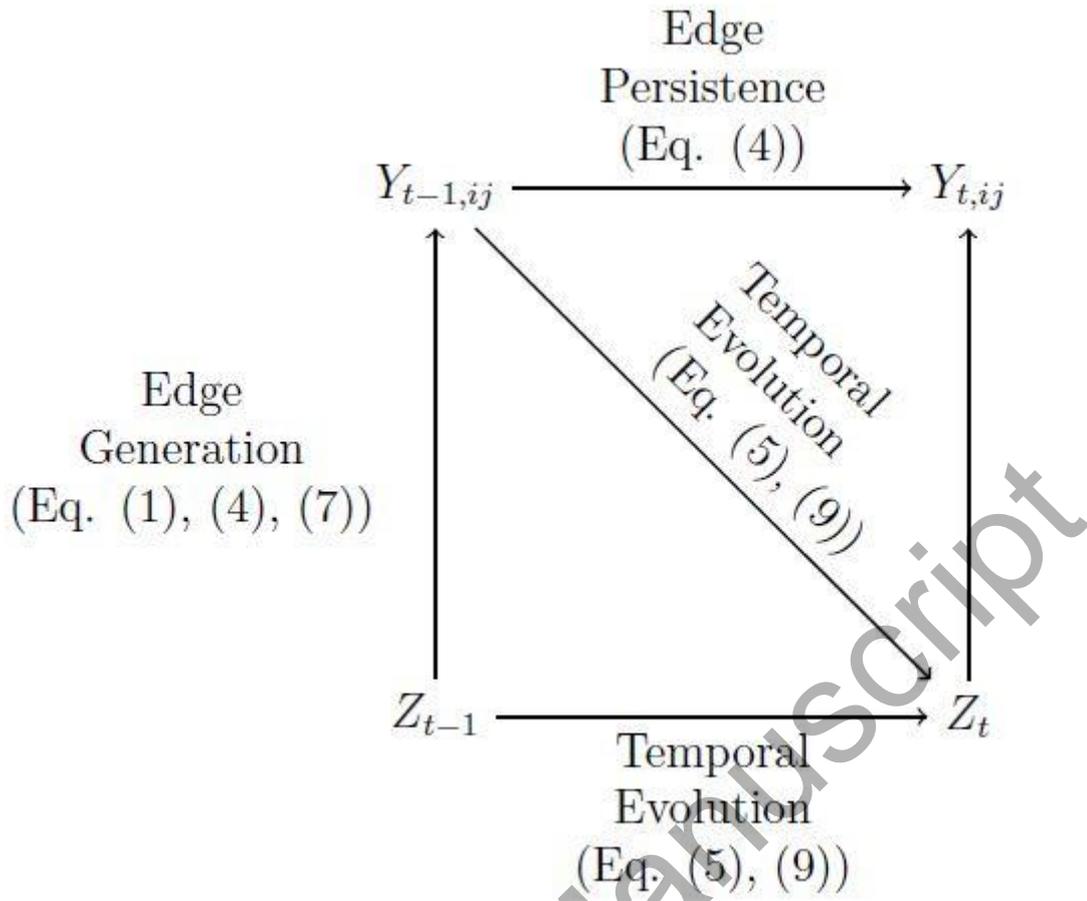
$$Y_{t-1,ij} \xrightarrow{\text{Edge Persistence (Eq. (4))}} Y_{t,ij}$$

Edge Persistence (Eq. (4))

$Y_{t-1,ij}$ → $Y_{t,ij}$

Edge Generation (Eq. (1), (4), (7))

Temporal Evolution (Eq. (5), (9))

Z_{t-1} → Z_t

Temporal Evolution (Eq. (5), (9))

Figure 3: Illustration of the Variance Estimation Method in a Bivariate Distribution

Edge Densities of Twitter Congressional Hashtag Networks

— Among Democrats
--- Among Republicans
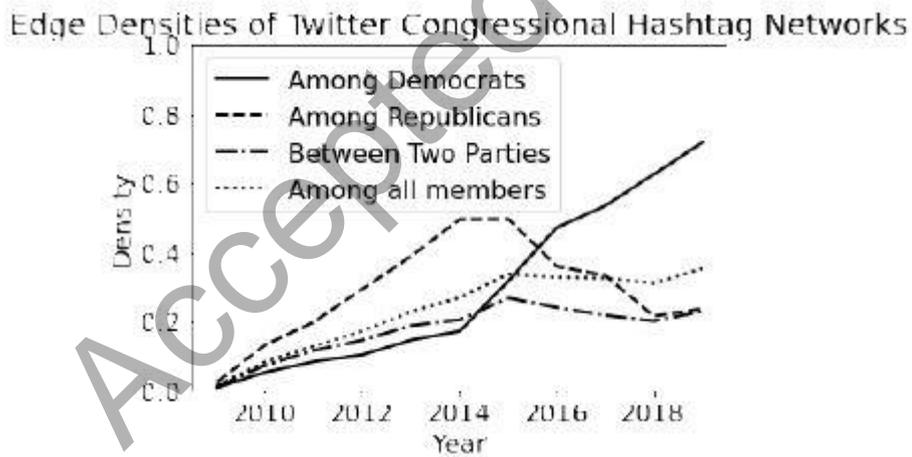-·- Between Two Parties
···· Among all members

Figure 4: Temporal evolution of edge density within the Democratic and Republican parties, the inter-party edge density and the edge density among all members
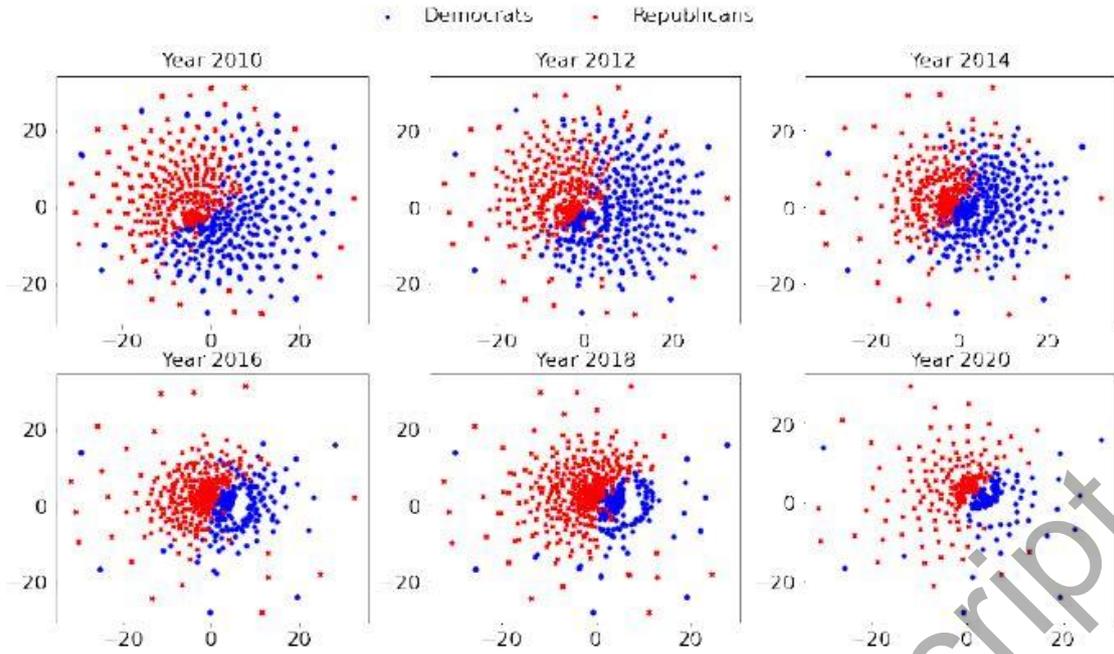
Figure 5: The point estimates of the latent positions with the extended model. The dots that mainly occupy the left portion of each plot represent Democrats and the dots that mainly occupy the right portion of each plot represent Republicans. At each time point, the two parties consistently occupy different halves of the space. The Democrats exhibited clustering behavior throughout the period. Conversely, the Republicans initially displayed a similar flocking behavior, but gradually began to disperse around the year 2016.
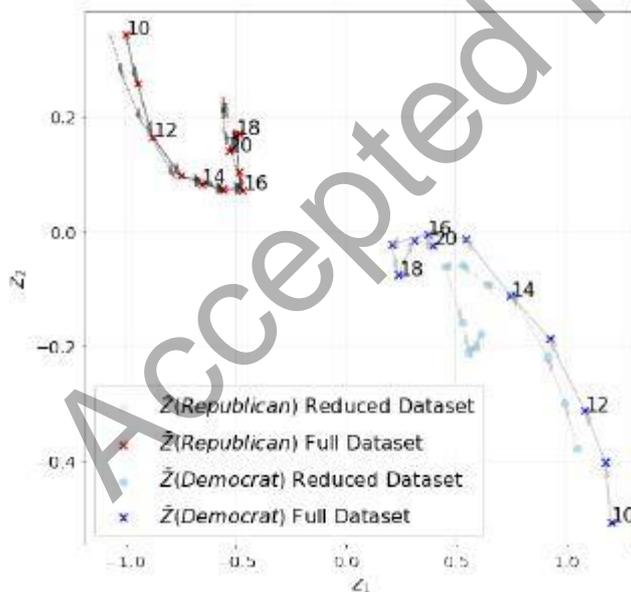

Figure 6: The trajectory of the mean of the latent positions of the members in each party. The average position of the two parties gradually and consistently approached each other from 2011 to 2016 and after 2016, both of them took a U-turn and moved apart.
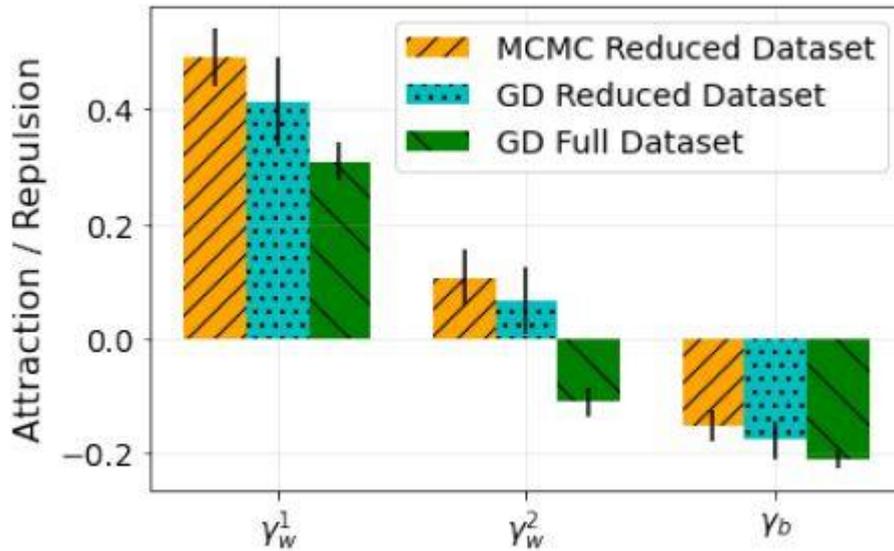
Figure 7: Comparison of attraction parameters from MCMC method on the reduced data set and the SGD-based method on the reduced and full data set. The MCMC method and the SGD-based method yield very similar point estimates and variance estimates for the attraction parameters on the reduced data set. On the full data set, the model concludes that there is a large attractive force within the Democratic Party. By contrast, we find that within the Republican Party the force is repulsive.
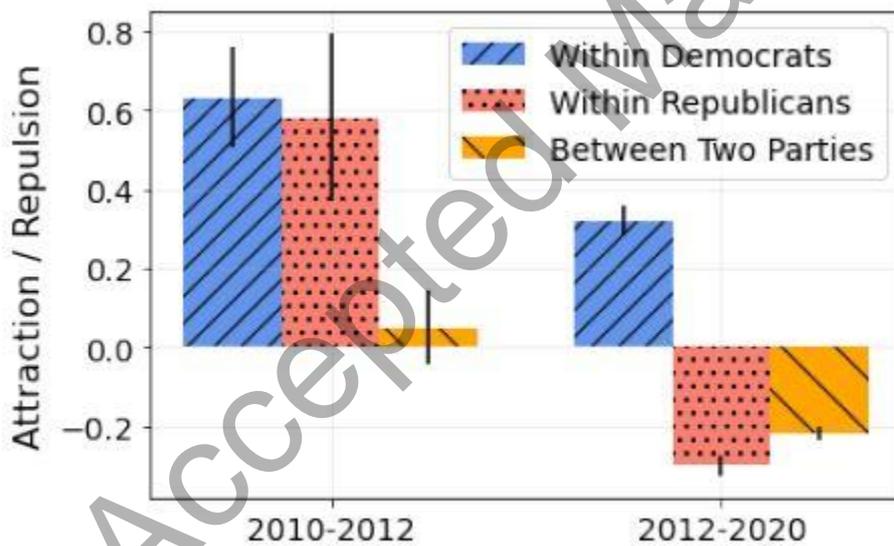

Figure 8: Evolution of posterior means and 95% CI for Democrats and Republicans, and between-group attraction/repulsion in X Congressional Hashtag Networks, the values of which at each time period in the horizontal axis are obtained by fitting the model using separately parameterized networks in the corresponding time periods.

Table 1: Posterior-based mean (empirical standard deviation) for point estimation (Algorithm 2) and variance estimation (Algorithm 3) in flocking and polarization settings, based on $n = 100$ and $n = 1000$ nodes, with $T = 10$ time points.

	$\alpha = 1$		$\delta = 2$		$\gamma^w = 0.25$		$\gamma^b = 0.5$	
	$\hat{\alpha}$	$\mathrm{Var}(\alpha)$	$\hat{\delta}$	$\mathrm{Var}(\delta)$	γ^w	$\mathrm{Var}(\gamma^w)$	γ^b	$\mathrm{Var}(\gamma^b)$
n=100	0.817 (0.029)	0.025 (0.001)	1.934 (0.023)	0.024 (0.001)	0.202 (0.135)	0.125 (0.026)	0.489 (0.14)	0.117 (0.011)
n=1000	0.972 (0.003)	0.003 (< 0.001)	1.992 (0.003)	0.002 (< 0.001)	0.25 (0.028)	0.03 (0.001)	0.498 (0.030)	0.028 (0.001)
	$\alpha = 1$		$\delta = 3$		$\gamma^w = 0.45$		$\gamma^b = -0.5$	
	$\hat{\alpha}$	$\mathrm{Var}(\alpha)$	$\hat{\delta}$	$\mathrm{Var}(\delta)$	γ^w	$\mathrm{Var}(\gamma^w)$	γ^b	$\mathrm{Var}(\gamma^b)$
n=100	0.825 (0.043)	0.038 (0.001)	2.868 (0.045)	0.038 (0.001)	0.302 (0.041)	0.035 (0.003)	-0.54 (0.035)	0.03 (0.004)
n=1000	0.971 (0.003)	0.004 (< 0.001)	2.976 (0.003)	0.004 (< 0.001)	0.433 (0.018)	0.015 (0.004)	-0.509 (0.017)	0.014 (0.004)

Table 2: 1-Democrats, 2-Republicans. The point estimates and the posterior standard deviation of the global parameters in the extended model fitted to the full X data set spanning from 2010 to 2020. The between-group attraction is -0.22, indicating that there is polarization between the two parties. The within group coefficient is 0.34 for Democrats, and -0.11 for Republicans, indicating that the Democrats were flocking, while the Republicans were polarizing within their own group.

	$\hat{\alpha}$	$\hat{\delta}$	$\hat{\gamma}_1^w$	$\hat{\gamma}_2^w$	$\hat{\gamma}^b$
Mean	3.2	1.08	0.34	-0.11	-0.22
SD	0.011	0.010	0.017	0.013	0.0085