On Dynamic Network Models and Application to Causal Impact

Yu-Chia Chen
yuchaz@uw.edu
University of Washington
Seattle, Washington

Avleen S. Bijral
avbijral@microsoft.com
Microsoft
Redmond, Washington

Juan Lavista Ferres
jlavista@microsoft.com
Microsoft
Redmond, Washington

ABSTRACT

Dynamic extensions of Stochastic block model (SBM) are of importance in several fields that generate temporal interaction data. These models, besides producing compact and interpretable network representations, can be useful in applications such as link prediction or network forecasting. In this paper we present a conditional pseudo-likelihood based extension to dynamic SBM that can be efficiently estimated by optimizing a regularized objective. Our formulation leads to a highly scalable approach that can handle very large networks, even with millions of nodes. We also extend our formalism to causal impact for networks that allows us to quantify the impact of external events on a time dependent sequence of networks. We support our work with extensive results on both synthetic and real networks.

CCS CONCEPTS

• **Computing methodologies** → **Unsupervised learning**; **Maximum a posteriori modeling**; • **Mathematics of computing** → *Stochastic processes.*

KEYWORDS

dynamic networks, stochastic block model, clustering, causal impact

ACM Reference Format:
Yu-Chia Chen, Avleen S. Bijral, and Juan Lavista Ferres. 2019. On Dynamic Network Models and Application to Causal Impact. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330990

1 INTRODUCTION

Network analysis is now a staple in many domains including computer science, social sciences and statistics. The main driver behind this growth has been the ubiquity of connection data, e.g., in the form of social, ecological or biological interactions. These interactions when modeled as edges with the entities as nodes naturally lead to a network representation.

One of the more prominent tasks in network analysis is the study of more compact representation of networks to aid in various applications. The model based approaches include random graph models [13], precision matrix based method [1, 2, 50] or embedding based methods [14, 18, 38]. A popular generative representation of a network is a Stochastic block model (SBM) [20]. In this approach edges are drawn i.i.d in Bernoulli trial with probability P_{c_i,c_j} for blocks or communities $c_i, c_j \in \{1, \ldots K\}$ that nodes i and j belong to. Given the observed unweighted adjacency matrix $\mathbf{A}$, the expected values of an edge given the labels $\mathbf{c}$ is $\mathbb{E}[A_{ij}|\mathbf{c}] = P_{c_i,c_j}$. This model captures the intuition that nodes in the same block share similar connectivity. A more general Degree-corrected SBM (DC-SBM) [24] was proposed to deal with degree heterogeneity wherein $\mathbb{E}[A_{ij} \mid \mathbf{c}] = \vartheta_i \vartheta_j P_{c_i,c_j}$, with an identifiability constraint $\sum_i \vartheta_i \mathbb{1}(c_i = l) = 1 \; \forall \; l \in \{1, \cdots, K\}$. Several methods exist to estimate static SBM – spectral [22, 31], Bayesian approaches include [2, 13, 33]. Alternatively, profile likelihood [7] and method of moments [8] have also been proposed.

More recently, models for time dependent networks have appeared in the literature [4, 15, 30, 46, 48]. Besides standard network modeling applications many other problems can be modeled as dynamic networks (e.g., face interaction [5]). The underlying intuition behind these methods is that the evolution of networks captures information about changes in the structure that the static model cannot. One can then extend the idea of identifying communities or blocks and model changes in the dynamic setting. This is of great importance in various applications such as link prediction [23, 40] or future network forecasting [26]. However a central challenge in applications is often the scale of the networks – **each** network in the sequence can be of the order of tens of millions of nodes. A fundamental challenge in estimating block models is that optimizing over label assignments is NP-hard and most existing work consists of Bayesian models that depend on intensive sampling based inference. This difficulty is further compounded when there are several time-dependent networks. For static SBM some recent methods [3, 17, 39] can scale up to a few million nodes but in the dynamic setting we are not aware of methods that work for very large networks.

In this paper we develop a highly scalable dynamic extension to a static profile likelihood (SPL) method [3] that works well for even sparse networks. This extension is enabled by modeling the parameters of the SPL method as stochastic processes, which can then be estimated in an efficient regularization framework. Moreover, we present a novel extension of our framework to model and estimate significant structural changes that can occur in networks due to events. This can be of great importance in trying to understand causal external influence on network generation dynamics. For e.g. in the Enron email network analysis presented in Xu and Hero [47], the publicizing of the financial scandal caused email traffic to peak between the CEO and company presidents and as such, one may be interested in quantifying the impact of this exogenous effect on the network dynamics. This perspective can also be thought of

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330990

as an extension of the causal impact methodology for time series introduced in Box and Tiao [9] to a network sequence.

1.1 Related Work

Dynamic network models has been around for a little over a decade or so and some of the early works were focused on latent variable models [41, 42]. The latent variable approach, while useful, is harder to interpret and does not usually scale well to very large scale networks. See Kim et al. [25] for an extensive survey. Consequently, dynamic SBM approaches have appeared in the literature. A state space model for dynamic SBM was proposed in Yang et al. [48], wherein the block membership for nodes is modeled as a latent variable generated by a Markov process. The authors propose a MCMC technique for estimation but this approach does not scale to even moderately large networks. In Xu and Hero [47], the authors propose a modification of the Yang et al. [48] model in that, instead of modeling the block membership directly, the authors propose a linear dynamic model for the logit transformed block transition probabilities. This model is then estimated using an extended Kalman filter and a local search method to align block membership across time steps (roughly speaking). This approach scales as $O(|E^t| + K^6 + |V^t|K^5 l)$ where $|V^t|$ is the number of nodes, K is the number of blocks and l is the number of local search iterations. Xu [46] extended the method to allow for the absence/presence of an edge across/within blocks at time t to depend on a previous state. From a model perspective this is an improvement since it captures the persistence/absence of edges across time compared to the possibly unrealistic assumption of conditional independence of all past networks given the current state. However, computationally both Xu [46], Xu and Hero [47] have a prohibitive cost even for moderately large networks. In contrast, our method scales well due to multiple reasons, a) the SPL approach compresses the adjacency matrix into row sums corresponding to blocks and efficiently optimizes a much simpler likelihood model, b) our relabeling step (for label alignment across time) scales as $O(K^3)$ which is significantly faster than existing approaches and c) our implementation depends on stochastic gradient descent which gives us a considerable advantage over MCMC based approaches.

More recently, Bao and Michailidis [4] proposed a community detection algorithm for time dependent networks that decomposes the graph adjacency sequence into a low rank, sparse and a noise component. These are estimated using an optimization problem that enforces sparsity and penalizes large changes in the low rank components over time. This approach is similar in spirit to ours, in that we also penalize large changes in the network parameterization but the optimization problem in Bao and Michailidis [4] is much more computationally intensive. On the theoretical front some results for dynamic network estimation appeared in Pensky [36], Pensky and Zhang [37].

1.2 Contribution

Our main contributions in this paper can be summarized as follows

(1) We propose a novel dynamic pseudo-likelihood extension to stochastic block models for networks. The estimation is posed as a regularized optimization problem which is solved using an efficient stochastic gradient descent (sgd) approach.
(2) Unlike existing work, our approach is highly scalable and works for very large networks. We are able to work with networks with tens of millions of nodes.
(3) We also present a novel framework for causal impact that extends the classical time series based intervention analysis methodology [9, 10] to networks. This allows us to separate exogenous artifacts of impact of events from the intrinsic network sequence generation model.
(4) Finally we show empirical results on several synthetic and real networks of different sizes.

2 ALGORITHM

2.1 Notation and Problem Formulation

Throughout this paper, matrices and vectors are denoted using bold letters. The super script denotes the matrices/vectors at the t-th snap shot, e.g., the t-th adjacency matrix to be $\mathbf{A}^t$. A collection of all matrices/vectors across the time is denoted as $\mathbf{A}^{1:T}$. For simplicity, $[K]$ represents the set $\{1, \cdots, K\}$. A more extensive table of notation is provided in Table 3 in the appendix. In the interest of completeness we first propose a problem statement

Definition 1 (Problem formulation). *Given a sequences of graphs $G^t(V^t, E^t)\ \forall\ t \in [T]$, with the corresponding un-weighted adjacency matrices $\mathbf{A}^{1:T}$ and vertex size (fixed over time), $|V^t| = N\ \forall\ t$. Let K be the number of communities in the graph and assume that the sequence of graphs is generated from a stochastic block model with underlying communities $\mathbf{c}^{1:T}$ with $\mathbf{c}^t \in [K]^N\ \forall\ t$, probability matrices $\mathbf{P}^{1:T}$ and degree heterogeneity parameters $\boldsymbol{\vartheta}^{1:T}$. We are interested in estimating the underlying communities $\hat{\mathbf{c}}^{1:T}$ and the evolution of edge probability matrices $\hat{\mathbf{P}}^{1:T}$ over time.*

For networks with different vertices sizes across time, one can always create a new series of graph with fixed vertices size $N = \max_t |V^t|$ by adding isolated nodes. Throughout the paper. we will use the term community and block interchangeably.

2.2 Background

The proposed dynamic pseudo-likelihood (DPL) algorithm extends the conditional pseudo-likelihood (SPL) approach for DC-SBM proposed in Amini et al. [3]. Given a static unweighted graph $G(V, E)$ of node size $|V| = N$ with $\mathbf{A} \in \mathbb{R}^{N \times N}$ as the unweighted adjacency matrix, the SPL method simplifies the combinatorial estimation of a block model by modeling instead the pseudo-observation *block sums* $\mathbf{B}(e) = \mathbf{A1}(e) \in \mathbb{R}^{N \times K}$ computed using an initialization of the community membership $\mathbf{e}$. Here $\mathbf{1}(e)$ is a $0-1$ label indicator matrix, a $N \times K$ matrix where the il-th entry is 1 if the initial node i belongs to community l. Note that in Amini et al. [3], many heuristics have been proposed to obtain such initialization of labels $\mathbf{e}$. Throughout the paper, we used the spectral clustering with perturbation to obtain the initial labels. Count vector $\mathbf{n}(\mathbf{e})$ and a count matrix $\mathbf{S}(\mathbf{e})$, represent the number of nodes of community l and the number of edges between block l and k, respectively, can be defined as: $[\mathbf{n}(\mathbf{e})]_l = \sum_i 1(e_i = l)$ and $[\mathbf{S}(\mathbf{e})]_{lk} = n_l(e)n_k(e)$ for $l \neq k$ and $[\mathbf{S}(\mathbf{e})]_{ll} = n_l(e)(n_l(e) - 1)$. The *maximum likelihood estimator* (MLE) for probability matrix $\mathbf{P}$ [24] is $\mathbf{P}(\mathbf{e}) = (\mathbf{1}(e)^T \mathbf{B}(e)) \oslash \mathbf{S}(e)$ with $\oslash$ represents the Hadamard (element-wise) division of two matrices.

The MLE for degree parameter ϑ is $[\boldsymbol{\vartheta}(e)]_i \propto d_i$ with an identifiably constraint $\sum_i \vartheta_i 1(c_i = l) = 1 \; \forall \, l \in [K]$.

Given the degree parameter $\boldsymbol{\vartheta}$, the model assumes that block sums of node i, $\mathbf{B}_i$ are generated from a mixture of multinomial distributions, with the probability of membership in community l is π_l. Given the community label l of node i, $\mathbf{B}_i$ is sampled from a multinomial distribution where the total number of trials is ϑ_i and event probability vector is $\boldsymbol{\theta}_l$. The likelihood function is then (1).

$$\ell(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathbf{B}) = \sum_{i=1}^{N} \log \left(\sum_{l=1}^{K} \pi_l \exp \left(\sum_{q=1}^{K} b_{iq} \log \theta_{lq} \right) \right) \tag{1}$$

Because of the simplicity of the likelihood function, we have a closed form update of EM algorithm. Therefore, the optimal $\boldsymbol{\pi}, \boldsymbol{\Theta}$ parameter given an initial guess of the communities structure $\mathbf{e}$ can be obtained by several EM steps until convergence. Upon obtaining the MLE of $\boldsymbol{\pi}$ and $\boldsymbol{\Theta}$, one can update the community structure by the posterior distribution as in (2).

$$\psi_{il} = \Pr(c_i = l | \mathbf{b}_i, \boldsymbol{\pi}, \boldsymbol{\Theta}) = \frac{\pi_l \exp(\sum_q b_{iq} \log \theta_{lq})}{\sum_k \pi_k \exp(\sum_q b_{iq} \log \theta_{lq})} \tag{2}$$

The maximizer class of the posterior distribution of node i is the updated block which i belongs to. With the updated block membership, a new pseudo-observation *Block sums* $\mathbf{B}$ can thus be obtained. The above procedure is repeated for M outer iterations and the class maximizer of the posterior distribution in the last is the output of the algorithm. The author proposed to initialize the parameters as in (3). Here $\boldsymbol{\Lambda}$ is a K by K matrix with its kl entry represents the expected number of edges between community k and l. The multinomial parameter $\boldsymbol{\Theta}$ is the row-normalization of the aforementioned edge propensity parameter $\boldsymbol{\Lambda}$.

$$\begin{gathered} \boldsymbol{\pi} = \mathbf{n}/N \,, \; \boldsymbol{\Lambda} = \mathrm{diag}(\mathbf{n})\mathbf{P} \,, \; \boldsymbol{\Theta} = \mathbf{D}_{\boldsymbol{\Lambda}}^{-1} \boldsymbol{\Lambda} \\ \text{where } \left[\mathrm{diag}(\mathbf{D}_{\boldsymbol{\Lambda}}^{-1})\right]_l = \textstyle\sum_k \lambda_{lk} \end{gathered} \tag{3}$$

The algorithm is summarized in Algorithm 1. For more details the reader can refer to Amini et al. [3].

Algorithm 1: Conditional Pseudo-likelihood algorithm

```
CPL_static (Λ, e, M) ⇒ Return: ĉ, Θ̂, π̂
Set ĉ_0 ⟵ e, initialize π, Λ using eq. (3) with e
for m = 1 : M do
    Update block sums B with estimated labels ĉ_{m−1}
    while not converge do
        E Step: Compute posterior Ψ using eq. (2)
        M step: π̂_l = Σ_i ψ_il / N;  θ̂_lk = Σ_i ψ_il b_ik / Σ_i ψ_il d_i
    end
    Update the labels: [ĉ_m]_i = arg max_k ψ_ik
end
```

2.3 Dynamic pseudo-likelihood estimation.

In the static pseudo-likelihood model, the two parameters $\boldsymbol{\pi}$ and $\boldsymbol{\Lambda}$ govern the underlying structure of the observed network (i.e., the block sums). A natural dynamic extension of such a model is to propose stochastic processes that capture some underlying intuition of how the networks evolve. We set random walk priors on $\boldsymbol{\pi}$ and $\boldsymbol{\Lambda}$, with the assumption that the parameters and hence the underlying networks vary slowly over time. We will relax this assumption for more variability in the parameter evolution in Section 3. Note that we have the probability simplex constraint on $\boldsymbol{\pi}$ and $\boldsymbol{\Lambda}$ is entry wise positive. For $\boldsymbol{\pi}$, we introduce a hidden parameter $\boldsymbol{\zeta}$ which is a unconstrained parameter with the following prior

$$\Pr\left(\zeta_l^t \mid \zeta_l^{t-1}; \sigma_1\right) = \mathcal{N}\left(\zeta_l^t; \zeta_l^{t-1}, \sigma_1\right) \; \forall \, l \in [K] \tag{4}$$

$\boldsymbol{\pi}$ can be obtained by a SoftMax transformation of $\boldsymbol{\zeta}$, i.e., $\boldsymbol{\pi} = \mathrm{SoftMax}(\boldsymbol{\zeta})$. $\boldsymbol{\Lambda}$ is assumed to follow the truncated normal process

$$\Pr\left(\lambda_{lk}^t \mid \lambda_{lk}^{t-1}; \sigma_2\right) = \mathcal{N}_T\left(\lambda_{lk}^t; \lambda_{lk}^{t-1}, \sigma_2, 0, \infty\right) \; \forall \, l, k \in [K] \tag{5}$$

The graphical representation of parameters is shown in Figure 1. The posterior distribution of the parameters $\boldsymbol{\pi}^{1:T}$ and $\boldsymbol{\Lambda}^{1:T}$ is

$$\begin{aligned} \Pr(\boldsymbol{\pi}^{1:T}, \boldsymbol{\Lambda}^{1:T} | \mathbf{B}^{1:T}) &\propto \Pr(\mathbf{B}^{1:T} | \boldsymbol{\pi}^{1:T}, \boldsymbol{\Lambda}^{1:T}) \cdot \Pr(\boldsymbol{\pi}^{1:T}) \cdot \Pr(\boldsymbol{\Lambda}^{1:T}) \\ &= \prod_{t=1}^{T} \Pr(\mathbf{B}^t | \boldsymbol{\pi}^t, \boldsymbol{\Lambda}^t) \cdot \Pr(\boldsymbol{\pi}^t | \boldsymbol{\pi}^{t-1}) \cdot \Pr(\boldsymbol{\Lambda}^t | \boldsymbol{\Lambda}^{t-1}) \end{aligned}$$

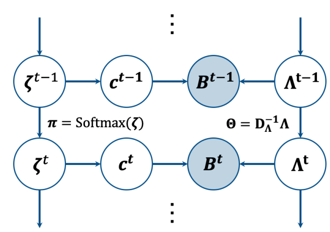


Figure 1: Graphical representation of the proposed DPL model.

The maximum-a-posteriori (MAP) estimator $\hat{\boldsymbol{\pi}}^{1:T}, \hat{\boldsymbol{\Lambda}}^{1:T}$ is the minimizer of the following negative log likelihood objective (6), with $\chi_1 = \frac{1}{2\sigma_1^2}$ and $\chi_2 = \frac{1}{2\sigma_2^2}$. Note that since the regularizers are the inverse variances of the prior processes one can use intuition about the network changes to come up with a reasonable search range. In our implementation we use projected stochastic gradient descent (SGD) to solve the optimization problem (6), with the proposed DPL algorithm summarized in Algorithm 2.

$$\begin{aligned} f(\boldsymbol{\zeta}^{1:T}, \boldsymbol{\Lambda}^{1:T} | \boldsymbol{B}^{1:T}; \chi_1, \chi_2) = &- \sum_{t=1}^{T} \ell(\boldsymbol{\pi}^t, \boldsymbol{\Theta}^t; \boldsymbol{B}^t) \\ &+ \chi_1 \sum_{t=2}^{T} \|\boldsymbol{\zeta}^t - \boldsymbol{\zeta}^{t-1}\|^2 \\ &+ \chi_2 \sum_{t=2}^{T} \|\boldsymbol{\Lambda}^t - \boldsymbol{\Lambda}^{t-1}\|_F^2 \end{aligned} \tag{6}$$

s.t. $\lambda_{lk} \geq 0$ for all l, k

with ℓ from eq. (1), $\boldsymbol{\pi} = \mathrm{SoftMax}(\boldsymbol{\zeta})$ and Θ from eq. (3).

Algorithm 2: Dynamic Pseudo-likelihood algorithm

DPL_estimate ($\mathbf{A}^{1:T}, \mathbf{e}^{1:T}, M, \chi_1, \chi_2$)
  Return: $\hat{\mathbf{c}}^{1:T}, \hat{\mathbf{\Lambda}}^{1:T}, \hat{\boldsymbol{\pi}}^{1:T}$
Set $\hat{\mathbf{c}}_0^{1:T} = \mathbf{e}^{1:T}$
for $m = 1 : M$ **do**
  Initialize $\boldsymbol{\pi}^{1:T}, \mathbf{\Lambda}^{1:T}$ with $\hat{\mathbf{c}}_{m-1}^{1:T}$ using eq. (3)
  Update $\mathbf{B}^{1:T}$ with $\hat{\mathbf{c}}_{m-1}^{1:T}$
  $\hat{\boldsymbol{\zeta}}^{1:T}, \hat{\mathbf{\Lambda}}^{1:T} = \arg\min f(\boldsymbol{\zeta}^{1:T}, \mathbf{\Lambda}^{1:T} | \mathbf{B}^{1:T}; \chi_1, \chi_2)$ in eq. (6)
  Compute posterior $\mathbf{\Psi}^{1:T}$ using eq. (2)
  $\mathbf{\Psi}^{1:T}$ = relabel_with_posterior($\mathbf{\Psi}^{1:T}$)
    ▷ algorithm 3
  Update the labels: $[\hat{\mathbf{c}}_m^t]_i = \arg\max_k \psi_{ik}^t$
end
$\hat{\mathbf{c}}^{1:T} = \hat{\mathbf{c}}_M^{1:T}$; Update parameter $\hat{\boldsymbol{\pi}}^{1:T}, \hat{\mathbf{\Lambda}}^{1:T}$ with eq. (3)

2.4 Relabel with posterior

Static community detection algorithms are only identifiable up to a permutation. For a temporal model, permutation between different time steps will result in class mismatch, which can create mapping issues between the parameters in the proposed model. In the dynamic community detection framework, there are two types of permutations, which are defined as follow.

Definition 2 (Local and global permutation). *For a ground truth label sequence $\mathbf{c}^{1:T}$, there is a global permutation in a given label sequence $\mathbf{e}^{1:T}$ if for all $i \in [N]$ and $t \in [T]$, there exists a one to one mapping $v_g : [K] \to [K]$ such that $c_i^t = v_g(e_i^t)$. Local permutations in a given label $\mathbf{e}^{1:T}$ exist if for all $i \in [N]$, there exists a one to one mapping $v^t : [K] \to [K]$ for time t such that $c_i^t = v^t(e_i^t)$.*

The aforementioned mapping is an issue when there is *local permutation* given the estimated communities. *Global permutations*, in contrast, will not create such issue. This motivates us to investigate an approximately identifiable model up to *global permutations*. To handle the the mapping issue, we propose to relabel the communities at each outer iteration using the posterior probability $\mathbf{\Psi}$. The method is inspired by Stephens [43] and essentially finds the permutation mapping v^t between time step $t-1$ and t by solving a Bayes risk minimization problem

$$v_*^t = \arg\min_v \sum_{i=1}^{N} \mathrm{KL}\left(\psi_{i,v(k)}^{t-1} \,\|\, \psi_{i,k}^t\right) \qquad (7)$$

The objective aims at finding the best local mapping v_*^t between the labels at time $t-1$ and t by aligning the posterior distribution. Because we are interested in having an approximately identifiable model up to *global permutations*, without loss of generality, we can set the global mapping v_g^1 to be identity. The global mapping at time t can be updated by the mapping of time t using $v_g^t(k) = v_*^t(v_g^{t-1}(k))$.

The minimizer of the objective as in (7) can be obtained in polynomial time by casting it to an assignment problem. Consider two posterior distribution $\mathbf{\Psi}^{t-1}$ and $\mathbf{\Psi}^t$ at time $t-1$ and t, one can construct a cost matrix $\mathbf{C}$ with $C_{lk} = \sum_i \mathrm{KL}\left(\psi_{il}^{t-1} \,\|\, \psi_{ik}^t\right)$. From this, Equation (7) can be rewritten as

$$\min_x \sum_{l,k} C_{lk} x_{lk} \text{ s.t. } \sum_l x_{lk} = 1; \sum_k x_{lk} = 1 \qquad (8)$$

Where the solution $x \in \{0,1\}^{K \times K}$ is a binary K by K matrix with non zero terms in the minimizer x_{lk}^* correspond to $k = v_*^t(l)$. The constraints in (8) are needed to ensure the solution to be a one to one mapping v^t from $[K]$ to $[K]$. Equation (8) is a classical form of an assignment problem, which can be solved in $O(K^3)$ time using Hungarian algorithm. A summary of the proposed relabeling procedure can be found in Algorithm 3.

Algorithm 3: Relabel with posterior

relabel_with_posterior ($\mathbf{\Psi}^{1:T}$)
  Return: Aligned posterior $\mathbf{\Psi}_*^{1:T}$
for $t = 2 : T$ **do**
  $v_*^t = \arg\min_v \sum_i^N \mathrm{KL}\left(\psi_{i,v(k)}^{t-1} \,\|\, \psi_{i,k}^t\right)$
  Global mapping $v_g^t(k) = v_*^t(v_g^{t-1}(k))$
    ▷ v_g^1 is identity
  for $k = 1 : K$ **do**
    $[\mathbf{\Psi}_*^t]_{\cdot,k} = \mathbf{\Psi}^t_{\cdot, v_g^t(k)}$
  end
end

3 CAUSAL IMPACT ON NETWORKS

On March 17, 2018 the story about the Facebook-Cambridge Analytica data harvesting scandal came to light [11]. Subsequently, a quit Facebook movement [21] gained momentum and a question of interest could be the impact of this story on Facebook's social network. In general, a way to quantify the impact of an external event on networks is relevant in many applications especially where running a randomized experiment is not feasible. Please see [6, 19, 44] for some relevant literature on experimentation in networks. Other related tasks like estimating causal impact of peer influence in networks [45] or change point detection in networks data [12, 35] have also been studied. Unlike these methods, we are interested in modeling *out* the impact of the exogenous event from the network generation dynamics. To the best of our knowledge, we are not aware of work that tackles this problem.

A simple way to inspect impact could be to compute network metrics over time [29]. One problem with this approach is that its not clear what the appropriate metric is and secondly global metrics may not capture local variations at all. For instance, consider a network where one community gets more dense and another gets sparse, the net effect might remain the same after an event.

We instead propose to merge community detection with impact assessment by taking a cue from the classical intervention analysis methodology described in Box and Tiao [9]. The idea here is to propose a model for the impact of the event and estimate it on top of the generative model of the network sequence. We achieve this by surmising and then estimating a model for the impact by proposing a modification to the stochastic process describing the parameters of the DPL model (2). This is necessary since the original model assumes a smooth transition for the parameters, i.e., (4) and (5), and hence may be insufficient to capture different transitions. More importantly our approach is also interpretable, since the parameters of our model have an intuitive meaning, e.g., the $\boldsymbol{\Lambda}$ matrix captures the intensity of the edge formation and an external event could alter it due to nodes dropping in or out in ways we explore next.

Note that this approach depends on the dynamic block model assumption in definition 1 and any violations can alter the assessment of the exogenous impact. It maybe feasible to test the model for goodness-of-fit as is done in Lei et al. [28] in the static setting. We leave this for future work.

3.1 DPL Extensions

In this section we explore two impact models and describe their estimation procedures.

3.1.1 Level Shift Estimation. First we extend the DPL model to the case where an external event at $t = T_0$ causes a level jump/drop in the edge formation intensity. In this modification, the random walk prior remains across time step except when $t = T_0$, when a level increase or decrease might occur. The joint distribution can therefore be written in the following form:

$$\Pr\left(\boldsymbol{\Lambda}^{1:T}\right) = \prod_{t \neq T_0} \Pr\left(\boldsymbol{\Lambda}^t \mid \boldsymbol{\Lambda}^{t-1}\right) \tag{9}$$

Plugging this into the likelihood function results in taking out a $\chi_2 \|\boldsymbol{\Lambda}^{T_0} - \boldsymbol{\Lambda}^{T_0-1}\|_F^2$ term from the DPL objective as in (6).

3.1.2 Drift Estimation. In the previous model, we assume that the edge intensity across time is a random walk without drift. However, it could be the case that the edge propensity has an increasing or decreasing trend which in turn could be affected by an external event, leading to a change in direction or magnitude. In this case, drift terms in the random walk model are needed to fully capture the evolution of the networks. One can model a random walk with drift with the following prior distribution. Here we illustrate the idea by adding drift term onto diagonal terms only. Therefore, the parameters to be estimated is a vector with length K. However, it is also possible to assume a drift on the off-diagonal terms.

$$\Pr\left(\lambda_{ll}^t \mid \lambda_{ll}^{t-1}; \sigma_2, \delta_l\right) = \mathcal{N}_T\left(\lambda_{ll}^t; \lambda_{ll}^{t-1} + \delta_l, \sigma_2, 0, \infty\right) \quad \forall\, l \in [K] \tag{10}$$

After combining the above two extensions of the model, one can write down the following joint distribution of $\boldsymbol{\Lambda}^{1:T}$, with the drift term before T_0 to be $\boldsymbol{\delta}_1$ while the drift term after T_0 to be $\boldsymbol{\delta}_2$.

$$\Pr\left(\boldsymbol{\Lambda}^{1:T}\right) = \prod_{t=2}^{T_0-1} \Pr\left(\boldsymbol{\Lambda}^t \mid \boldsymbol{\Lambda}^{t-1}; \sigma_1, \boldsymbol{\delta}_1\right) \prod_{t=T_0+1}^{T} \Pr\left(\boldsymbol{\Lambda}^t \mid \boldsymbol{\Lambda}^{t-1}; \sigma_2, \boldsymbol{\delta}_2\right)$$

With the above modification, the objective shown in (6) becomes

$$f(\boldsymbol{\zeta}^{1:T}, \boldsymbol{\Lambda}^{1:T} | \boldsymbol{B}^{1:T}; \chi_1, \chi_2) = -\sum_{t=1}^{T} \ell(\boldsymbol{\pi}^t, \Theta^t; \boldsymbol{B}^t) + \chi_1 \sum_{t=2}^{T} \|\boldsymbol{\zeta}^t - \boldsymbol{\zeta}^{t-1}\|^2$$
$$+ \chi_2 \sum_{t=2}^{T_0-1} \|\boldsymbol{\Lambda}^t - \boldsymbol{\Lambda}^{t-1} - \mathrm{diag}(\boldsymbol{\delta}_1)\|_F^2 + \chi_2 \sum_{t=T_0+1}^{T} \|\boldsymbol{\Lambda}^t - \boldsymbol{\Lambda}^{t-1} - \mathrm{diag}(\boldsymbol{\delta}_2)\|_F^2$$
$$\text{s.t. } \lambda_{lk} \geq 0 \text{ for all } l, k; \text{ with } \boldsymbol{\pi} = \mathrm{SoftMax}(\boldsymbol{\zeta}) \text{ and } \Theta \text{ from eq. (3).} \tag{11}$$

For purely *level shift* model, we set $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ to be the zero vector. As for purely *drift* model, the second to the last summation will end at $t = T_0$ rather than $t = T_0 - 1$. Note that, we can extend our approach to more complex impact terms but for the purpose of exposition the two models are sufficient.

3.2 Impact Assessment and Re-sampling

Once we have an estimate of the intervention parameters we need to assess their significance and that requires uncertainty bands $\mathcal{B}_\alpha$ of the *counterfactual* model $\mathcal{W}$ with significance level α. This allows us to determine whether the change of the network parameters are significant. Based on the proposed generative DPL model, we propose a resample procedure to estimate the uncertainty band $\mathcal{B}_\alpha$ by repeatedly generating J resamples of series of networks corresponding to *impact model* $\mathcal{I}$ and the *counterfactual model* $\mathcal{W}$. A point estimate of the edge probability matrices of model $\mathcal{I}$ can be obtained by the average of the resampled data, while the uncertainty band of model $\mathcal{W}$ can be constructed from the $\alpha/2$-th to $1-\alpha/2$-th quantile of the empirical distribution of the resampled data. Details of resampling procedure can be found in Section A.

4 EXPERIMENTS

In this section we provide empirical support for our work and experiment with different datasets including (1) synthetic dataset with 8000 nodes `Synth8000`, (2) MIT reality mining dataset [16] `MITReal` and (3) MathOverflow dataset [34] `MathO`. For analysis of causal impact, we explore a modified version of `MITReal` and Enron email dataset [27] `EnronMail`.

For evaluation we consider local and global clustering metrics. Local metrics evaluate the community structure at each time separately and global ones evaluate the clustering for all the time steps. When the ground truth labels are available we use normalized mutual information (NMI) score and for the more realistic case we use modularity score [32] to compare competing community structures. These scores essentially measure the difference between the fraction of edges connected between nodes of same groups and the expected fraction if edges are distributed randomly. Moreover, since the modularity score is consistent under the DC-SBM model [49], it serves as a good criteria for evaluating competing models for networks with no ground truth labels.

Two different global metrics are used in evaluating the estimated labels when the ground truth labels are available. First one is called *global normalized mutual information* (global NMI), which evaluates the NMI scores of N nodes across all time steps together. Note that local NMI scores are invariant under local permutations, while global NMI scores only under global permutations. We also use precision/recall/F1 scores for classifying nodes that switch classes over time. Since our model is a dynamic extension of SPL we refer the reader to the section 4 of Amini et al. [3] for runtime comparisons. Their experiments reveal that the runtime for the static method is comparable to spectral clustering and over 10 time faster than belief propagation. We compare our method (DPL) to the following approaches (a) Proposed Algorithm 2 without relabeling (sDPL) (b) Static pseudo likelihood (SPL) approach proposed in Amini et al.

[3] (each time step estimated independently) (c) Spectral clustering with perturbation (SPC) of Amini et al. [3] and (d) Dynamic stochastic block model (DSBM) of Xu and Hero [47].

4.1 Synthetic dataset – `Synth8000`

The synthetic dataset `Synth8000` used in the experiment consists of $N = 8,000$ nodes, with communities size $K = 30$ and total time steps $T = 60$, with average degree of the networks to be around 20 for each time step. The details of generating the dataset can be found in Section B.1. The top row of Figure 2 shows the local metrics of the synthetic dataset. Purple line in Figure 2a and 2b show the result of the proposed DPL method. On average, the proposed algorithm has a 3% gain in NMI scores across all time compared to the result of SPL. Figure 2b shows the modularity scores of the communities estimated by different models over time. The figure indicates that the estimated labels of DPL model are almost as good as ground truth labels (blue line) in terms of modularity score. Figure 2c shows the 2D histogram of each class versus time. The y-axis of each subfigure represents different communities, while x-axis represents the time. The color in each cell represents the frequency of occurrence, with lighter color represents higher frequency. The ground truth 2D histogram in the bottom-right subfigure of 2c, one can show a smooth transition of the estimated class labels across time. DPL and sDPL both show some kind of smooth transition, however, this is not the case for SPL as shown in the bottom-left panel of the figure.

Global metrics of different algorithms are shown in Table 1. The proposed algorithm, DPL, has the best global performance. sDPL, which is the dynamic version of psuedo-likelihood without relabeling, shows a gain in all global metrics. These results indicate the importance of relabeling for improved performance in dynamic stochastic block models.

For both change class detection and global NMI scores, DPL has the best performance compared to other methods. Although only have a 3% gain in performance in local metric, we have a nearly 20% gain in the global NMI scores with the time dependent model. As expected, SPL and SPC does not perform well in terms of global metrics, for all time steps are modeled independently. sDPL takes into account the time dependency of smooth transition of model, which results in a drastic gain in the global metrics. However, since relabeling step is not used, we see a loss of performance in the class switching prediction problem.

Table 1: Global metrics for `Synth8000` dataset.

	precision	recall	F1	NMI
DPL	0.134	0.983	**0.236**	**0.744**
sDPL	0.107	0.986	0.194	0.552
SPL	0.040	0.969	0.077	0.013
SPC	0.040	0.967	0.077	0.009

We were unable to run DSBM for this model as the implementation provided to us does not scale for community size $K = 30$ (see Section 1.1). Additionally, since dropping the relabeling step from DPL results in a drop in results, we exclude any further comparison with it.

4.2 MIT reality mining dataset – `MITReal`

In the MIT reality mining dataset [16] we have $T = 37$ snapshots of cell phone activity between $N = 94$ students and staffs at MIT. Each snapshot corresponds to one week of data. The preprocessing method used in this work is similar to that developed in Xu and Hero [47], which construct the temporal networks by physical proximity. The affiliation of participants were known, which corresponds to *students* and *faculties and staffs* (so $K = 2$), was used as the ground truth label in the experiment. The proximity data we used were between August 2004 till May 2005. Note that in this dataset, ground truth labels do not change with time. However, the propensities of edge formation change drastically with time because of different events in the academic calendars (e.g., exam week, winter break, etc.), making it suitable for a dynamic SBM.

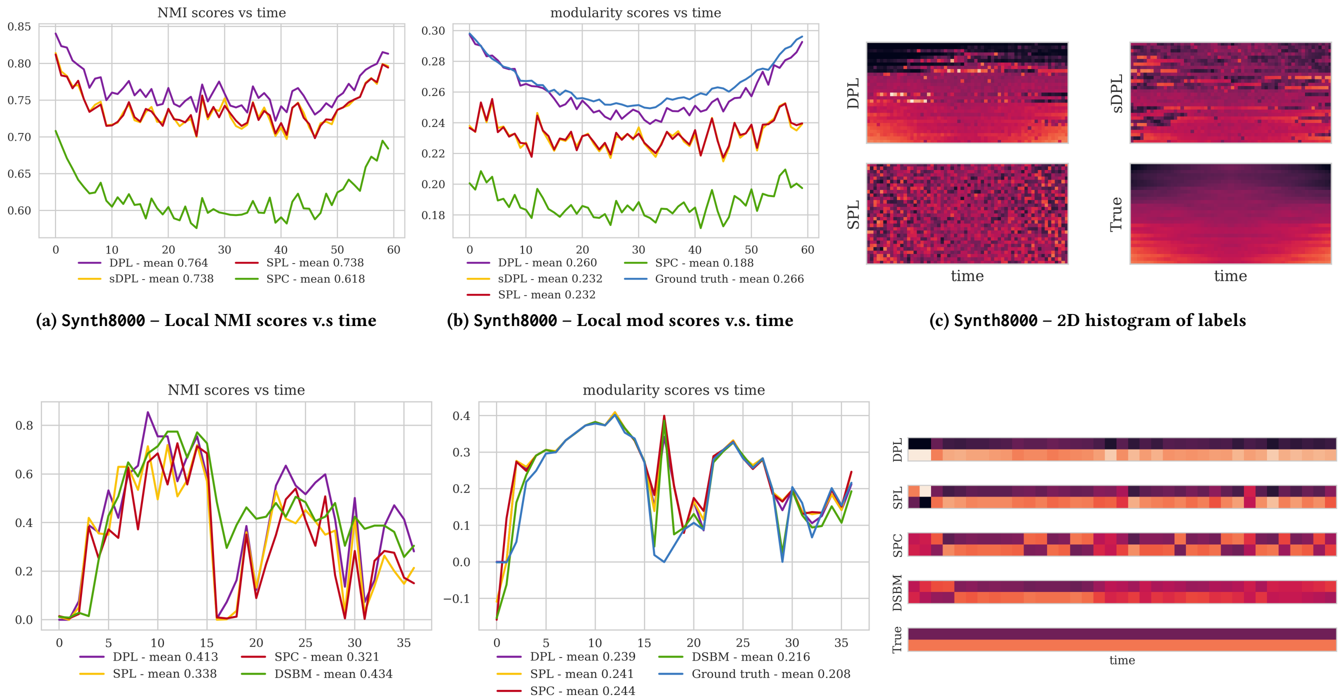
The second row of Figure 2 shows the local metrics for different inference algorithms. The drop in the range of $t \in [16, 18]$ and $t \in [31, 32]$ correspond to the winter and spring break when the networks get sparse. In the local metrics, DSBM performs slightly better than the proposed algorithm DPL, since the local search mechanism in the DSBM model searches for labels for each node directly as opposed to aligning the posteriors (like our approach). However, Figure 2f reveals that it does not do as well as DPL in aligning clusters. The communities are consistent and the transitions are smooth across all time in DPL, while the estimated labels of other models suffer from class switching. This also affect the results of global metrics as in Table 2, with DPL has the best global NMI scores among all the considered algorithms.

Table 2: Global NMI score for `MITReal`

	DPL	SPL	SPC	DSBM
Global NMI	**0.399**	0.327	0.032	0.377

4.3 MathOverflow dataset – `MathO`

MathOverflow dataset [34], `MathO`, is a temporal network dataset of the question and answer forum *MathOverflow*. The data collects the activities of $N = 24,818$ users across $2,350$ days. An edge at time t $e^t = (i, j)$ is formed if user i had answered or commented on user j's question or answer at time t. The temporal edges in the dataset is $506,550$. In our experiment, each snapshots of graphs correspond to one month of data, with total number of snap shots $T = 79$. We further turned the original directed graph into an undirected version to fit the proposed DPL model. The number of classes used in the experiment is $K = 15$, which approximately corresponds to the active research topics in mathematics. We compared our approach to two other community detection methods, SPC and SPL as we were unable to scale up to 24k nodes for the remaining methods. Figure 3 shows the modularity score at different time steps. DPL model has a 4% increase over SPL model in the modularity scores on average across all time. The gain in the performance is more significant at later time steps.



(a) `Synth8000` – Local NMI scores v.s time (b) `Synth8000` – Local mod scores v.s. time (c) `Synth8000` – 2D histogram of labels

(d) `MITReal` – Local NMI scores v.s time (e) `MITReal` – Local mod scores v.s. time (f) `MITReal` – 2D histogram of labels

Figure 2: Local metrics on `Synth8000` and `MITReal`. Left column contains figures with local NMI scores. Middle column is the modularity scores at each time steps. Right column is the 2D histogram of different models, with lighter colors represent cells with more counts. The first row is the experiment results for `Synth8000`, while the second row is the results of `MITReal`.

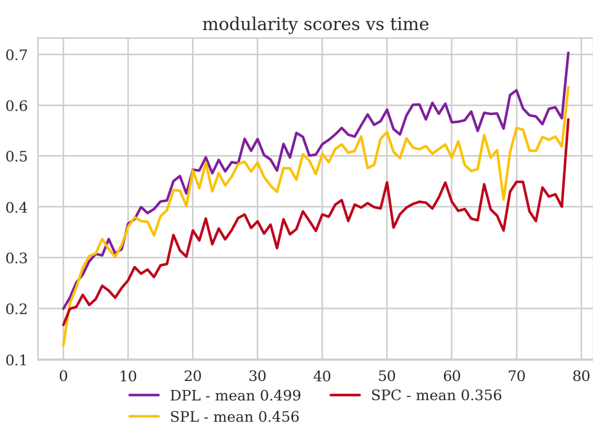


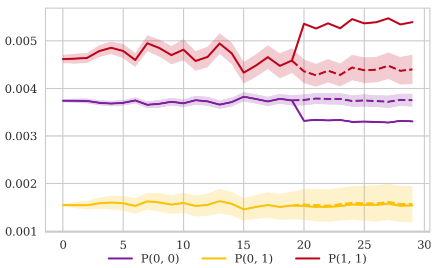
Figure 3: Modularity scores versus time for `Math0`.

4.4 Causal impact on network

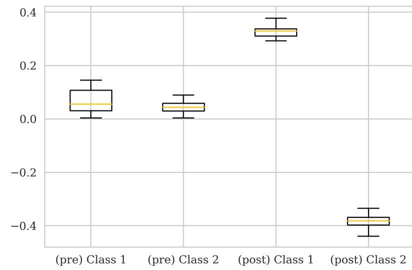
4.4.1 Synthetic dataset with level shift and drifts. For a simple demonstration we first create two synthetic datasets (`Synth8000Jump`, `Synth8000Drift`) with fixed $K = 2$ communities and $T = 30$. We inject changes at $T_0 = 20$ which leads to the larger community getting less and the smaller getting more dense. This change is introduced using a level shift and also a drift change in the edge formation probabilities of the two communities. See Section B.2 and B.3 for data generation details.

Figure 4 shows the result of the causal impact on network. The point estimate of the *impact* model $\mathcal{I}$ (solid line) and *counterfactual* model $\mathcal{W}$ (dash line) and the uncertainty bands are generated using the resampling procedure described in Algorithm 4 of Section A. Figure 4a shows the results for the `Synth8000Jump` dataset. The confidence bands reveal that we are able to recover the impact to the within-class edge probabilities, however the change to probability $\hat{P}_{01}$ is not significant. We see the same behavior for the drift interventions in Figure 4c. A well separation of same entries of drift terms $\boldsymbol{\delta}_1, \boldsymbol{\delta}_2$ before and after the impact shown in Figure 4b further confirm the significance of the impact has on the network.

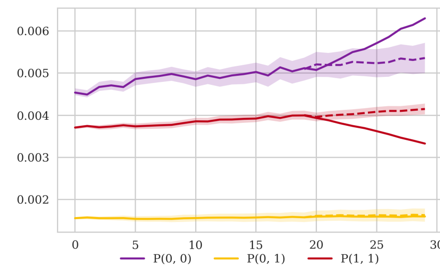
4.4.2 Fall semester MIT reality mining dataset. To test out approach on a real dataset we made modifications to the original `MITReal` described in Section 4.2 and refer to the new dataset as `MITRealFall`. In which, the first two time steps are discarded, since the fall semester started at the third week in `MITReal`. The Last week of instruction of fall quarter happened at $t = 14$, which is the time of intervention $T_0 = 14$ for our analysis. Final exams and winter breaks happened at $t \in [15, 20]$. Spring semester started at $t = 21$ therefore the networks after $t = 21$ are also discarded. The networks of interaction follow a steady pattern during the course instruction



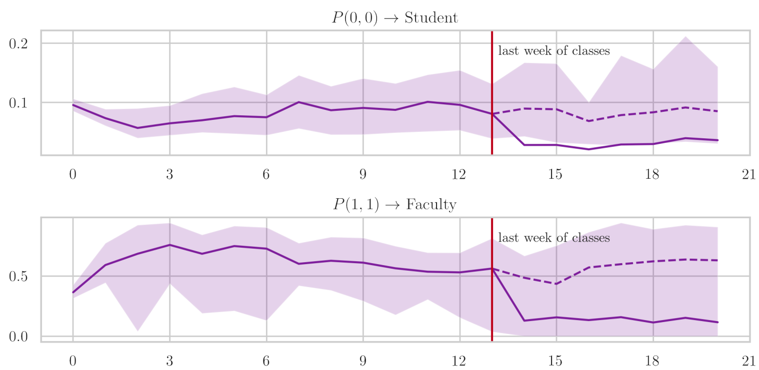
(a) `Synth8000Jump` – Estimated pmatrix



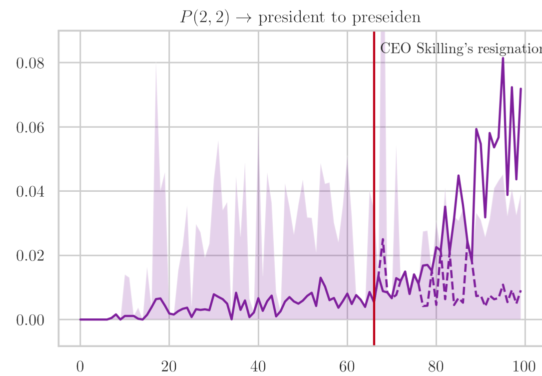
(b) `Synth8000Drift` – box plot of drifts δ_1, δ_2



(c) `Synth8000Drift` – Estimated pmatrix



(d) `MITRealFall` – estimated P matrix.



(e) `EnronMail` – estimated P matrix.

Figure 4: Results of causal impact on networks. Solid and dash lines represent the point estimate of *impact* model $P_{\mathcal{I}}$ and *counterfactual* model $P_{\mathcal{W}}$, respectively. The shaded region is the 5% uncertainty band $\mathcal{B}_{5\%}$. Figures 4a and 4c show the resampling results for `Synth8000Jump` and `Synth8000Drift`, respectively. Figure 4b is the box plot of drift vectors δ_1 and δ_2 for *impact model* $\mathcal{I}$ on `Synth8000Drift` dataset. Figures 4d and 4e show the the resampling results for `MITRealFall` and `EnronMail`, respectively.

period – which can be modeled with the DPL model (2). The intervention here then corresponds to the end of the term, the period after the last week of instruction. Note that the model can be easily modified to deal with multiple interventions.

Figure 4d shows the estimated $\hat{P}_{00}$ and $\hat{P}_{11}$ across the time span. Class 0 corresponds to the group of *students* while 1 corresponds to the group of *staffs and faculties*. Resampled estimator in the figure shows the drop in the edge propensity of students is significant, while for staff it is not. This is intuitive since the winter break is expected to have a more significant impact on the physical proximity between students compared to staffs and faculties.

4.4.3 Enron email dataset. The Enron dataset contains 4 years of email messages between 184 Enron employees from 1998 to 2002. There is an edge at time t if employee i and j communicated on email. Each snapshot consists of a week of data, which results in 189 time steps in the original graph. The dataset `EnronMail` used in the experiment contains the time steps from the 76th to the 176th weeks. Seven different roles of employees are known and served as the ground truth labels in the dataset. The roles of employees correspond to directors, CEOs, presidents, vice-presidents, managers, traders and others. The intervention (red line) corresponds to the event when CEO Skillings resigned in response to the publicized scandal. Figure 4e shows the estimated $\hat{P}_{22}$, which corresponds to the probability of forming edges between presidents. We can see that the estimated $\hat{P}_{22}$ is outside the 5% uncertainty band $\mathcal{B}_{5\%}$ of the counterfactual, which indicates that such event caused an "unnatural" change in the dynamics of email communication for a block of network nodes.

5 CONCLUSION

To fill the gap in the literature on large scale modeling of dynamic networks, we presented an efficient dynamic SBM framework (DPL). Our approach extends the static pseudo-likelihood approach of [3] to the time dependent case. The proposed regularization estimation framework is efficiently estimated using projected SGD and can scale up to tens of millions of nodes. Additionally, a novel extension of our approach to intervention analysis allows us to analyze artifacts of exogenous impact on networks. These artifacts correspond to structural changes in the intrepretable parameters of the DPL framework and are estimated with corresponding uncertainty quantile bands. For the practitioner this provides a useful tool to understand external influence on network sequences.

The theoretical time complexity per iteration for a time step is $O(|E_t| + NK^2)$. The $|E_t|$ term comes from the block sum computation $\mathbf{A^t 1(e)}$, which in the worst case is $O(N^2)$. In future work we plan to update the pre-computed block sum from a previous iteration using the proposed relabeling algorithm. We conjecture that this can reduce the time complexity to $O(K^3 + NK^2)$ per time step. We also hope to experiment with much larger networks in future work.

ACKNOWLEDGEMENT

We would like to thank Arash Amini for valuable discussions and Kevin Xu for sharing the preprocessed datasets.

REFERENCES

[1] Amr Ahmed and Eric P Xing. 2009. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences* 106, 29 (2009), 11878–11883.

[2] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. 2008. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research* 9, Sep (2008), 1981–2014.

[3] Arash A Amini, Aiyou Chen, Peter J Bickel, and Elizaveta Levina. 2013. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics* 41, 4 (2013), 2097–2122.

[4] Wei Bao and George Michailidis. 2018. Core community structure recovery and phase transition detection in temporally evolving networks. *Scientific reports* 8, 1 (2018), 12938.

[5] Alain Barrat and Ciro Cattuto. 2013. Temporal networks of face-to-face human interactions. In *Temporal Networks*. Springer, 191–216.

[6] Guillaume W Basse and Edoardo M Airoldi. 2018. Model-assisted design of experiments in the presence of network-correlated outcomes. *Biometrika* 105, 4 (2018), 849–858.

[7] Peter J Bickel and Aiyou Chen. 2009. A nonparametric view of network models and NewmanâĂŞGirvan and other modularities. *Proceedings of the National Academy of Sciences* (2009), pnas. 0907096106.

[8] Peter J Bickel, Aiyou Chen, and Elizaveta Levina. 2011. The method of moments and degree distributions for network models. *The Annals of Statistics* 39, 5 (2011), 2280–2301.

[9] George EP Box and George C Tiao. 1975. Intervention analysis with applications to economic and environmental problems. *Journal of the American Statistical association* 70, 349 (1975), 70–79.

[10] Kay H. Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L. Scott. 2015. Inferring causal impact using Bayesian structural time-series models. *Ann. Appl. Stat.* 9, 1 (03 2015), 247–274. https://doi.org/10.1214/14-AOAS788

[11] Carole Cadwalladr and Emma Graham-Harrison. 2018. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach . https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election.

[12] Simon De Ridder, Benjamin Vandermarliere, and Jan Ryckebusch. 2016. Detection and localization of change points in temporal networks with the aid of stochastic block models. *Journal of Statistical Mechanics: Theory and Experiment* 2016, 11 (2016), 113302.

[13] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. 2011. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E* 84, 6 (2011), 66106.

[14] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*. 3844–3852.

[15] Daniele Durante, David B Dunson, et al. 2016. Locally adaptive dynamic networks. *The Annals of Applied Statistics* 10, 4 (2016), 2203–2232.

[16] Nathan Eagle and Alex Pentland. 2006. Reality mining: sensing complex social systems. *Personal and ubiquitous computing* 10, 4 (2006), 255–268.

[17] Prem K Gopalan and David M Blei. 2013. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences* (2013), 201221839.

[18] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[19] Huan Gui, Ya Xu, Anmol Bhasin, and Jiawei Han. 2015. Network a/b testing: From sampling to estimation. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 399–409.

[20] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social networks* 5, 2 (1983), 109–137.

[21] Tiffany Hsu. 2018. For Many Facebook Users, a 'Last Straw' That Led Them to Quit . https://www.nytimes.com/2018/03/21/technology/users-abandon-facebook.html.

[22] Jiashun Jin et al. 2015. Fast community detection by SCORE. *The Annals of Statistics* 43, 1 (2015), 57–89.

[23] Ruthwik R Junuthula, Kevin S Xu, and Vijay K Devabhaktuni. 2016. Evaluating link prediction accuracy in dynamic networks with added and removed edges. In *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on*. IEEE, 377–384.

[24] Brian Karrer and Mark E J Newman. 2011. Stochastic blockmodels and community structure in networks. *Physical review E* 83, 1 (2011), 16107.

[25] Bomin Kim, Kevin H. Lee, Lingzhou Xue, and Xiaoyue Niu. 2018. A review of dynamic network models with latent variables. *Statist. Surv.* 12 (2018), 105–135. https://doi.org/10.1214/18-SS121

[26] Myunghwan Kim and Jure Leskovec. 2013. Nonparametric multi-group membership model for dynamic networks. In *Advances in neural information processing systems*. 1385–1393.

[27] Bryan Klimt and Yiming Yang. 2004. Introducing the Enron Corpus.. In *CEAS*.

[28] Jing Lei et al. 2016. A goodness-of-fit test for stochastic block models. *The Annals of Statistics* 44, 1 (2016), 401–424.

[29] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 177–187.

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

[31] Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103, 23 (2006), 8577–8582.

[32] Mark E J Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E* 69, 2 (2004), 26113.

[33] Krzysztof Nowicki and Tom A B Snijders. 2001. Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association* 96, 455 (2001), 1077–1087.

[34] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. 2017. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 601–610.

[35] Leto Peel and Aaron Clauset. 2015. Detecting change points in the large-scale structure of evolving networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[36] Marianna Pensky. 2016. Dynamic network models and graphon estimation. *arXiv preprint arXiv:1607.00673* (2016).

[37] Marianna Pensky and Teng Zhang. 2017. Spectral clustering in the dynamic stochastic block model. *arXiv preprint arXiv:1705.01204* (2017).

[38] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[39] Sandipan Roy, Yves Atchadé, and George Michailidis. 2018. Likelihood Inference for Large Scale Stochastic Blockmodels with Covariates based on a Divide-and-Conquer Parallelizable Algorithm with Communication. *Journal of Computational and Graphical Statistics* just-accepted (2018), 1–28.

[40] Purnamrita Sarkar, Deepayan Chakrabarti, and Michael Jordan. 2012. Nonparametric link prediction in dynamic networks. *arXiv preprint arXiv:1206.6394* (2012).

[41] Purnamrita Sarkar and Andrew W Moore. 2006. Dynamic social network analysis using latent space models. In *Advances in Neural Information Processing Systems*. 1145–1152.

[42] Daniel K Sewell and Yuguo Chen. 2015. Latent space models for dynamic networks. *J. Amer. Statist. Assoc.* 110, 512 (2015), 1646–1657.

[43] Matthew Stephens. 2000. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 62, 4 (2000), 795–809.

[44] Daniel L Sussman and Edoardo M Airoldi. 2017. Elements of estimation theory for causal effects in the presence of network interference. *arXiv preprint arXiv:1702.03578* (2017).

[45] Panos Toulis and Edward Kao. 2013. Estimation of causal peer influence effects. In *International conference on machine learning*. 1489–1497.

[46] Kevin Xu. 2015. Stochastic block transition models for dynamic networks. In *Artificial Intelligence and Statistics*. 1079–1087.

[47] Kevin S Xu and Alfred O Hero. 2014. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing* 8, 4 (2014), 552–562.

[48] Tianbao Yang, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin. 2011. Detecting communities and their evolutions in dynamic social networks-a Bayesian approach. *Machine learning* 82, 2 (2011), 157–189.

[49] Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. 2012. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics* 40, 4 (Aug. 2012), 2266–2292. https://doi.org/10.1214/12-AOS1036

[50] Shuheng Zhou, John Lafferty, and Larry Wasserman. 2010. Time varying undirected graphs. *Machine Learning* 80, 2-3 (2010), 295–319.

A DETAILS FOR RE-SAMPLING

The resampling procedure for obtaining the confidence band $\mathcal{B}_\alpha$ is summarized as follows: a model before the intervention T_0 was fit on a series of networks with adjacency matrices $\mathbf{A}^{1:T_0}$. χ_1 and χ_2 are chosen based on the metrics introduced in the experiment section. For real networks without ground truth label, modularity scores are used as the criteria. With the above regularization parameter, one can estimate the community $\hat{\mathbf{c}}^{1:T}$ and the underlying parameters $\hat{\boldsymbol{\Lambda}}^{1:T}$ of a the whole series of graph with adjacency matrices $\mathbf{A}^{1:T}$ with the modified DPL model as in (11).

To sample from the *impact* model $\mathcal{I}$, we first generate the j-th sample of edge propensity matrices $\boldsymbol{\Lambda}^{1:T}_{\mathcal{I},j}$ by the *random walk* prior with variance parameter $\sigma^2 = \frac{1}{2\chi_2^2}$. Here we fix the starting point $t = 1$ and change point $t = T_0$, which means $\boldsymbol{\Lambda}^t_{\mathcal{I},j} = \hat{\boldsymbol{\Lambda}}^t\ \forall\, t \in \{1, T_0\}$. Edge forming probability matrices can be computed and used in DC-SBM model to generate a series of graph $\mathbf{A}^{1:T}_{\mathcal{I},j}$. The modified DPL model as in (11) is then fit on the series of graph to estimate the j-th sample of probability matrix $\mathbf{P}^{1:T}_{\mathcal{I},j}$. After a total of J resamples, we obtain a collection of estimated probability matrices $\mathbf{P}^{1:T}_{\mathcal{I},1:J}$. Similar procedure is used in obtaining the estimates for *counterfactual* model $\mathcal{W}$, with only setting $\boldsymbol{\Lambda}^1_{\mathcal{W},j} = \hat{\boldsymbol{\Lambda}}^1$ rather than the change point. The series of graphs are then fitted with the normal DPL model as in Algorithm 2. An impact can be tested to be whether it is significant under the significant level α by testing whether $\hat{\mathbf{P}}^t_{\mathcal{I}} \in [\Phi^t_{\mathcal{W}}(\alpha/2), \Phi^t_{\mathcal{W}}(1-\alpha/2)]$ for $t \geq T_0$, where $\Phi^t_{\mathcal{W}}$ is the cumulative distribution function of $\mathbf{P}^t_{\mathcal{W}}$. In practice, we use the percentile from the resampled $\hat{\mathbf{P}}^t_{\mathcal{W},1:J}$ to form such uncertainty band and the average of $\hat{\mathbf{P}}^t_{\mathcal{I},1:J}$ to get the estimate of $\hat{\mathbf{P}}^t_{\mathcal{I}}$. Algorithm 4 in Section A summarized the aforementioned resampling procedure.

Algorithm 4: Re-sampling procedure

```
resampling (A^{1:T}, e^{1:T}, M)
Return: Collection of resamples P̂^{1:T}_{M,1:J}
  ▷ M is the model used, can be impact model I
    or counterfactual model W
Choose χ1, χ2 by Algorithm 2 with A^{1:T0}, e^{1:T0}.
  ▷ Select parameters before intervention T0.
Estimate ĉ^{1:T}, π̂^{1:T} and Λ̂^{1:T} with modified model as in (11).
Set the resampled parameters c_R^{1:T} = ĉ^{1:T} and Λ_R^1 = Λ̂^1
if M = I then
    Set Λ_R^{T0} = Λ̂^{T0}
end
Set σ_2^2 = 1/(2χ_2^2)
for j = 1 : J do
    if M = I then
        Resample Λ^{2:T0−1}_{I,j} and Λ^{T0+1:T}_{I,j} from (9) and (10).
    else
        Resample Λ^{2:T}_{W,j} from (5).
    end
    Obtain the MLE for P^{1:T}_{M,j} and ϑ^{1:T}_{M,j} as in Section 2.2.
    Generate adjacency matrix A^{1:T}_{M,j} with DC-SBM model.
    Estimate P̂^{1:T}_{M,j} and Λ̂^{1:T}_{M,j} by running Algorithm 2 with
      selected χ1, χ2 and outer iteration M = 1
end
P̂^{1:T}_{M,1:J} is the collection of P̂^{1:T}_{M,j} for j ∈ [J]
```

B DATA GENERATION

B.1 Synthetic dataset – Synth8000

The `Synth8000` dataset is generated as follows: $N = 8000$ nodes are randomly assigned to 30 communities with probability proportion to the size of the community at $t = 0$. The size of community k is $n_k = Nq_k$, with q_k drawn from a uniform distribution between $[0, 1]$ and normalized such that $\sum_k q_k = 1$. 10% of the node were selected to be the hub, with $\vartheta_{\text{hub}} = 5 \times \vartheta_{\text{ordinary}}$. The degree parameter of node i, ϑ_i is fixed across over time. Between $1 \leq t \leq 30$, 4% of the nodes will randomly switch their communities to other. At $31 \leq t \leq 60$, the process is reversed, with the communities been put back and return to original communities at $t = 60$.

With communities at time t, a random graph $G^t(V, E^t)$ is generated by the DC-SBM model. An edge $e^t = (i, j)$ of the graph G^t is formed according to the Bernoulli trial, with success probability $p = \vartheta_i \vartheta_j P_{c_i^t, c_j^t}$. The probability matrix $\mathbf{P} = (p_{\text{in}} - p_{\text{out}})\mathbf{I} + p_{\text{out}}\mathbf{1}\mathbf{1}^\top$, with in-class probability $p_{\text{in}} = 1.8 \times 10^{-3}$ and out-class probability $p_{\text{out}} = 1.6 \times 10^{-4}$.

B.2 Level shift dataset – Synth8000Jump

The community labels are constructed as follow: $N = 8000$ nodes are randomly assigned to one of the two communities, with the size of the larger community to have 70% of the nodes. 10% of the nodes are selected to be hubs with $\vartheta_{\text{hub}} = 5 \times \vartheta_{\text{ordinary}}$. The degree parameter $\boldsymbol{\vartheta}$ are fixed across all time. The first class correspond to the larger community.

The edge probability matrices $\mathbf{P}^{1:T}$ are generated as follow: first we set $\mathbf{P}^1 = (p^1_{\text{in}} - p^1_{\text{out}})\mathbf{I} + p^1_{\text{out}}\mathbf{1}\mathbf{1}^\top$ with $p^1_{\text{in}} = 0.004$ and $p^1_{\text{out}} = 0.0015$. The edge propensity matrix $\boldsymbol{\Lambda}^1$ is then calculated by (3). For $1 < t < T_0$ and $T_0 < t \leq T$, $\boldsymbol{\Lambda}^t$ is generated by (5). $\boldsymbol{\Lambda}^{T_0}$ is drawn from the same distribution but with mean parameter at $\boldsymbol{\Lambda}^{T_0-1} + \text{diag}(\boldsymbol{\eta})$, with $\boldsymbol{\eta} = [-3, 3]^\top$. From (3), we can calculate $\mathbf{P}^{1:T}$ from $\boldsymbol{\Lambda}^{1:T}$. A series of graph with adjacency matrices $\mathbf{A}^{1:T}$ can be generated by DC-SBM model with the calculated probability matrices $\mathbf{P}^{1:T}$.

B.3 Drift dataset – Synth8000Drift

The community labels are generated as the same way as mentioned in Section B.2 but with second class corresponds to the larger community. The probability matrix $\mathbf{P}^1$ and edge propensity matrix $\boldsymbol{\Lambda}^1$ at time $t = 1$ is generated using the same method as `Synth8000Jump` dataset. For $1 < t \leq T$, $\boldsymbol{\Lambda}^t$ is generated by the *random walk* prior with drift as in (10), with drift before intervention $\boldsymbol{\delta}_1 = [0.1, 0.1]^\top$ and after intervention $\boldsymbol{\delta}_2 = [0.5, -0.5]$. Probability matrices $\mathbf{P}^{1:T}$ are then calculated and are used to generated a series a graph using DC-SBM model.

C NOTATIONAL TABLE

The notational table can be found in Table 3.

Table 3: Notational table

Matrix operation	
$\mathbf{M}$	Matrix
$\mathbf{m}_i$	Column vector represents the i-th row of $\mathbf{M}$
$\mathbf{m}_{:,j}^T$	Column vector represents the j-th column of $\mathbf{M}$
m_{ij}	Scalar represents ij-th element of $\mathbf{M}$
$[\mathbf{M}]_{ij}$	Scalar, alternative notation for m_{ij}
$\mathbf{v}$	Column vector
v_i	Scalar represents i-th element of vector $\mathbf{v}$
$[\mathbf{v}]_i$	Scalar, alternative notation for v_i
$\mathbf{M}^t$	Matrix $\mathbf{M}$ at time t
$\mathbf{M}^{t_1:t_2}$	Collections of matrices $\mathbf{M}$ from $t = t_1$ to $t = t_2$
$\mathbf{M}_{R,j}^{t_1:t_2}$	The j-th resample of collection $\mathbf{M}^{t_1:t_2}$
$\mathbf{M}_{R,1:J}^{t_1:t_2}$	All the J resamples of collection $\mathbf{M}^{t_1:t_2}$
$\mathbf{M}_1 \oslash \mathbf{M}_2$	Hadamard (element-wise) division of $\mathbf{M}_1$ and $\mathbf{M}_2$
Scalars	
N	Vertices size
K	Number of blocks
T	Total time steps
χ_1	Regularizer corresponds to the evolution of $\boldsymbol{\pi}$
χ_2	Regularizer corresponds to the evolution of $\boldsymbol{\Lambda}$
M	Number of outer iterations
T_0	Time when the intervention happened
J	Number of resamples
Matrices	
$\mathbf{A}$	Adjacency matrix
$\mathbf{c}$	Truth community labels
$\mathbf{e}$	Initial guess of community labels
$\mathbf{P}$	Block probability matrix of DC-SBM
$\boldsymbol{\vartheta}$	Degree heterogeneity parameters for DC-SBM
$\mathbf{B}(\mathbf{e})$	Block sum matrix for a given label $\mathbf{e}$
$\mathbf{n}(\mathbf{e})$	Count array for a given label $\mathbf{e}$
$\mathbf{S}(\mathbf{e})$	Count matrix for a given label $\mathbf{e}$
$\boldsymbol{\pi}$	Prior distribution of community labels
$\boldsymbol{\Lambda}$	Edge propensity matrix
$\boldsymbol{\Theta}$	(Row) Normalized edge propensity matrix
$\boldsymbol{\zeta}$	Unconstraint parameter governs the transition of $\boldsymbol{\pi}$
$\boldsymbol{\Psi}$	Class posterior matrix
Miscellaneous	
$G^t(V^t, E^t)$	Graph with vertex set V^t and edge set E^t at time t
$\mathrm{KL}(P\|Q)$	KL divergence of distribution P and Q
ν^t	Local permutation of time t
ν_g	Global permutation
$\Phi_X(\alpha)$	Cumulative distribution function of X at level α
$[U]$	A set $\{1 \cdots, U\}$ with U be a positive integer
$\mathcal{W}$	Counterfactual model
$\mathcal{I}$	Impact model