UNIVERSITY OF CALIFORNIA
Los Angeles

Generative Model for Change Point Detection in Dynamic Weighted Graphs

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Junpeng Ren

2024

ABSTRACT OF THE THESIS

Generative Model for Change Point Detection
in Dynamic Weighted Graphs

by

Junpeng Ren
Master of Science in Statistics
University of California, Los Angeles, 2024
Professor Oscar Hernan Madrid Padilla, Chair

Existing methods for graph change point detection focus on binary graphs, limiting their ability to capture the richer information available in weighted graphs. This work introduces a novel algorithm for detecting change points in weighted graphs by combining generative modeling with Group-Fused Lasso. Graphs are represented through low-dimensional latent vectors, with neural networks modeling their underlying distributions. Specifically, change points are identified as shifts in the distributions of latent vectors, with Group-Fused Lasso optimized using the Alternating Direction Method of Multipliers (ADMM). Simulation experiments and real-world applications demonstrate the effectiveness of the proposed method, highlighting the advantages of incorporating edge weights for a deeper understanding of temporal graph dynamics.

The thesis of Junpeng Ren is approved.

Yik Lun Kei

Yingnian Wu

Guang Cheng

Oscar Hernan Madrid Padilla, Committee Chair

University of California, Los Angeles

2024

To all who dare to chase their dreams, may we find courage even in the face of storms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

My graduate journey has been an unforgettable chapter of my life, marking my first experience studying abroad. UCLA became a nurturing home where I grew immensely, both personally and academically. Here, I had the privilege of connecting with exceptional mentors, lifelong friends, and my partner, all of whom played a pivotal role in shaping who I am today.

First and foremost, I wish to express my deepest gratitude to Professor Oscar Hernan Madrid Padilla, Professor Guang Cheng, Professor Minjeon Jeon and Professor Yik Lun Kei for their unwavering support and invaluable mentorship throughout my time at UCLA. Your guidance opened the door to the world of statistical research and encouraged me to pursue challenges I never thought I could overcome. Without your help, I would not have achieved the growth and accomplishments that define this journey. I am also deeply thankful to my thesis committee members, professors, and academic peers, whose shared knowledge and encouragement helped me grow into a capable researcher. Your generosity and support will remain a lifelong treasure.

I met my life partner here, someone with whom I wish to share my future. We laughed and cried together, creating cherished memories during the best years of our lives. To Chenxin, thank you for standing by my side and encouraging me through every setback along the way. I could not have achieved what I have today without you.

Lastly, I owe my deepest gratitude to my family for their unconditional love and support. Your belief in me has been my anchor through every challenge, and I am forever grateful for your unwavering strength.

To everyone who has been part of this incredible journey—thank you from the bottom of my heart.

CHAPTER 1

Introduction

Graphs, also referred to as networks, are a fundamental data structure used to represent complex relationships between entities, where nodes represent the entities and edges capture their interactions. They are widely applied across numerous fields, including social science, neuroscience, and finance, among others. Capturing the evolution of graphs over time is a challenging problem, with change point detection being a critical yet complex task in this context.

Various models have been developed to represent graphs, such as the stochastic block model [HLL83], exponential random graphical models (ERGM) [LKR13], exponential random dot product graphs [YS07; Nic08], etc. Corresponding methods for change point detection have also been proposed for these models. For instance, the stochastic block model captures community structures within graphs by grouping nodes and defining edge probabilities based on group membership. [BBM20] introduces a method for detecting temporal changes in these community structures. Similarly, ERGMs model graphs as an exponential function of specified graph statistics. To account for temporal dynamics, Separable Temporal ERGMs [KH14] extend this framework to model evolving graph structures over time, with [KLCP23] offering insights into their change point detection. Random dot product graphs embed nodes in a latent space, with edge probabilities determined by the dot product of latent vectors, and corresponding change point detection focuses on shifts in these embeddings [LBFM21].

However, most existing methods for change point detection are designed for binary graphs, where edge weights are restricted to 0 or 1. The detection of change points in

weighted graphs, where edge weights carry richer information, has been less explored. Weighted graphs naturally encode more information than their binary counterparts. For example, in social graphs, edge weights can represent the strength of relationships between individuals, providing a more detailed and nuanced depiction of interactions [OP09].

In this work, we propose a novel algorithm for change point detection in weighted graphs, extending the existing binary approach presented in [KLLCP24]. Our approach leverages a generative model to simulate graphs and employs Group-Fused Lasso [TSRZK05] to localize change points. Specifically, we represent each graph with a low-dimensional latent random vector that captures the underlying graph-level distribution. Neural networks are then used to model the true distribution of these latent vectors. We define a graph's change point as the moment when the edge generation pattern changes, which corresponds to a shift in the distribution of the latent representation. By applying Group-Fused Lasso with Alternating Direction Method of Multipliers (ADMM, with partial lists of examples [BPCPE11], [FCG10], [BF10], etc.), we identify distributional shifts in the latent vectors and consequently detect the associated change points. This framework provides a principled and effective approach to uncovering changes in the evolution of weighted graphs.

CHAPTER 2

Background

In this chapter, we provide an overview of existing methods for modeling graph data and detecting change points in graphs, which serve as benchmarks for evaluating change point detection algorithms. Section 2.1 introduces the change point detection algorithm based on separable temporal exponential random graph models, an advanced approach for binary graph change point detection. Section 2.2 and 2.3 present two more general methods: graph-based change point detection [CZ15] and kernel-based change point detection [SC24]. While [CZ15; SC24] are not specifically designed for network data, these methods are applicable to weighted graphs, making them valuable baselines for weighted graph change point detection.

2.1 Change Point Detection Using Separable Temporal Exponential Random Graph Models

[KLCP23] introduces a change point detection algorithm, referred to as the CPD-STERGM method. This algorithm identifies change points in dynamic graphs by leveraging the Separable Temporal Exponential-family Random Graph Model (STERGM, [KH14]). Below, we provide a brief introduction to the key components of this algorithm.

2.1.1 Exponential Random Graph Models (ERGMs)

Exponential Random Graph Models (ERGMs, [LKR13]) are a flexible and widely used statistical framework for modeling complex networks. Consider a node set $N = \{1, 2, \ldots, n\}$

and a binary graph $\mathbf{y} \subseteq \mathcal{Y}$, where $\mathcal{Y} \subseteq N \times N$ represents all possible dyads (pairs of nodes). The dyadic variable y_{ij} equals 1 if a relationship exists between nodes i and j, and 0 otherwise. ERGMs assume that the probability of observing a specific graph $\mathbf{y}$ is determined by a set of graph-level statistics and their corresponding parameters. The probability mass function of an ERGM is given by:

$$\mathrm{P}(\mathbf{y}; \boldsymbol{\theta}) = \frac{\exp\left(\boldsymbol{\theta} \cdot \mathbf{g}(\mathbf{y})\right)}{\psi(\boldsymbol{\theta})},$$

where $\mathbf{g}(\mathbf{y})$ is a vector of sufficient statistics capturing graph properties (e.g., edge counts, triangle counts, degree distributions), $\boldsymbol{\theta}$ is the parameter vector associated with these statistics, and $\psi(\boldsymbol{\theta})$ is the normalizing constant (log-partition function) ensuring proper normalization:

$$\psi(\boldsymbol{\theta}) = \sum_{\mathbf{y}' \in \mathcal{Y}} \exp\left(\boldsymbol{\theta} \cdot \mathbf{g}(\mathbf{y}')\right).$$

The parameters $\boldsymbol{\theta}$ reflect the strength and direction of the effects of the corresponding graph statistics. A positive θ_k indicates that the associated statistic $g_k(\mathbf{y})$ is more likely to occur, while a negative θ_k suggests suppression of that statistic.

ERGMs are particularly suitable for static networks and are foundational for understanding network processes. However, they are limited when it comes to modeling temporal dynamics, which motivates the need for extensions like the Separable Temporal Exponential Random Graph Model (STERGM, [KH14]).

2.1.2 Separable Temporal Exponential Random Graph Models

The Separable Temporal Exponential-family Random Graph Model (STERGM, [KH14]) is an extension of the ERGM that allows for the modeling of dynamic networks by separating the formation and dissolution of ties. This separation provides a flexible framework to study temporal network evolution by decoupling the processes of creating and dissolving ties.

With the same notation and denoting the network at time t as $\mathbf{y}^t$, STERGM can be constructed from two intermediate networks: the formation network $\mathbf{y}^{+,t}$ and the dissolution network $\mathbf{y}^{-,t}$. Specifically:

$$y_{ij}^{+,t} = \max(y_{ij}^{t-1}, y_{ij}^t), \quad y_{ij}^{-,t} = \min(y_{ij}^{t-1}, y_{ij}^t).$$

Assuming conditional independence between $\mathbf{y}^{+,t}$ and $\mathbf{y}^{-,t}$ given $\mathbf{y}^{t-1}$, the probability of observing the network $\mathbf{y}^t$ can be expressed as:

$$\mathrm{P}(\mathbf{y}^t \mid \mathbf{y}^{t-1}; \boldsymbol{\theta}^t) = \mathrm{P}(\mathbf{y}^{+,t} \mid \mathbf{y}^{t-1}; \boldsymbol{\theta}^{+,t}) \times \mathrm{P}(\mathbf{y}^{-,t} \mid \mathbf{y}^{t-1}; \boldsymbol{\theta}^{-,t}),$$

where the formation and dissolution probabilities are modeled as:

$$\mathrm{P}(\mathbf{y}^{+,t} \mid \mathbf{y}^{t-1}; \boldsymbol{\theta}^{+,t}) = \exp[\boldsymbol{\theta}^{+,t} \cdot \mathbf{g}^+(\mathbf{y}^{+,t}, \mathbf{y}^{t-1}) - \psi(\boldsymbol{\theta}^{+,t}, \mathbf{y}^{t-1})],$$

$$\mathrm{P}(\mathbf{y}^{-,t} \mid \mathbf{y}^{t-1}; \boldsymbol{\theta}^{-,t}) = \exp[\boldsymbol{\theta}^{-,t} \cdot \mathbf{g}^-(\mathbf{y}^{-,t}, \mathbf{y}^{t-1}) - \psi(\boldsymbol{\theta}^{-,t}, \mathbf{y}^{t-1})].$$

Here, $\mathbf{g}^+$ and $\mathbf{g}^-$ are vectors of network statistics for formation and dissolution, and $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ are the respective parameter vectors. The term $\psi(\cdot)$ is the log-partition function ensuring proper normalization.

STERGM's flexibility makes it particularly suitable for analyzing temporal network dynamics, where the processes of tie creation and removal exhibit distinct patterns. However, it assumes that the parameters $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ remain constant over time, which may not hold in practice when the network undergoes structural changes.

2.1.3 Change Point Detection in STERGM

In dynamic networks, the underlying generative processes often change over time, resulting in structural shifts. Change point detection aims to identify these time points where significant changes occur in the network's structure or dynamics. Here, we introduce the

change point detection algorithm based on STERGM, the CPD-STERGM [KLCP23].

For a time-varying STERGM [KH14], the parameters $\boldsymbol{\theta}^t = (\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t})$ may vary across time. Denote the ordered change points as $\{C_k\}_{k=0}^{K+1} \subseteq \{1, 2, \ldots, T\}$, where $K+1$ segments are defined, and the parameters are constant within each segment:

$$\boldsymbol{\theta}^{C_k} = \boldsymbol{\theta}^{C_k+1} = \cdots = \boldsymbol{\theta}^{C_{k+1}-1}, \quad k = 0, \ldots, K,$$

but differ between segments:

$$\boldsymbol{\theta}^{C_k} \neq \boldsymbol{\theta}^{C_{k+1}}, \quad k = 0, \ldots, K-1.$$

To detect these change points, the negative log-likelihood of the STERGM is minimized with a Group Fused Lasso penalty:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -\ell(\boldsymbol{\theta}) + \lambda \sum_{t=2}^{T-1} \frac{\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^{t-1}\|_2}{w_t},$$

where $\lambda > 0$ is a regularization parameter and w_t are position-dependent weights. To identify change points, the differences between consecutive parameters are computed:

$$\Delta\hat{\boldsymbol{\theta}}^t = \|\hat{\boldsymbol{\theta}}^{t+1} - \hat{\boldsymbol{\theta}}^t\|_2, \quad t = 1, \ldots, T-1.$$

These differences are standardized to account for variability:

$$\Delta\hat{\zeta}^t = \frac{\Delta\hat{\boldsymbol{\theta}}^t - \text{median}(\Delta\hat{\boldsymbol{\theta}})}{\text{sd}(\Delta\hat{\boldsymbol{\theta}})}.$$

A data-driven threshold ϵ_{thr} is defined as:

$$\epsilon_{\text{thr}} = \text{mean}(\Delta\hat{\zeta}) + Z_{1-\alpha} \times \text{sd}(\Delta\hat{\zeta}),$$

where $Z_{1-\alpha}$ is the $(1-\alpha)$-quantile of the standard normal distribution. Time points t

with $\Delta\hat{\zeta}^t \geq \epsilon_{\text{thr}}$ are classified as change points.

This approach provides a robust framework for detecting change points in temporal networks, enabling the identification of significant shifts in the formation and dissolution dynamics modeled by STERGM.

2.2 Graph-based change-point detection

Graph-based change-point detection [CZ15] provides a flexible, nonparametric framework to detect distributional changes in a sequence of observations. By leveraging similarity graphs, this method is well-suited for complex data structures, such as high-dimensional data and non-Euclidean sample spaces. This section introduces the theoretical foundation, hypothesis testing framework, graph construction methods, and the mathematical formulation of the detection process.

2.2.1 Problem Setup

Given a sequence of observations $\{x_i : i = 1, 2, \ldots, n\}$, the goal is to detect one or more change-points where the distribution of the observations changes. Specifically, let the observations be indexed by i in a meaningful order, such as time or spatial location.

Single Change-Point Scenario Under the null hypothesis H_0, all observations are identically distributed with distribution F_0:

$$H_0 : x_i \sim F_0, \quad \forall i = 1, \ldots, n.$$

The alternative hypothesis H_1 assumes a single change-point τ such that:

$$H_1 : \begin{cases} x_i \sim F_0, & i \leq \tau, \\ x_i \sim F_1, & i > \tau, \end{cases}$$

where $F_0 \neq F_1$. The change-point τ is unknown and must be estimated.

To test such kind of hypothesis, a similarity graph G is needed to be constructed, which is specifically induced for change point detection in this setting, over the observations and use graph-based statistics to quantify group separation.

2.2.2 Graph Construction

The similarity graph G is defined by a vertex set corresponding to the observations $\{x_i\}$ and an edge set representing pairwise relationships. Graph-based hypothesis testing has been studied using various graph construction methods, including testing with the Minimum Spanning Tree (MST, [FR79]), Minimum Distance Pairing (MDP, [Ros05]), etc. We use the MST as an example to illustrate the procedure of constructing a similarity graph. It is worth noting that other graph construction methods are also integrated into the proposed framework.

Minimum Spanning Tree (MST): The MST connects all observations with the minimum total edge weight, where the weight is typically defined using a distance metric such as the Euclidean distance:

$$\text{Distance}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2},$$

where d is the d-th dimension of the i-th observed vector $\boldsymbol{x}$.

2.2.3 Graph-Based Statistic

Given a similarity graph G, the test statistic $R_G(t)$ quantifies the number of edges connecting nodes in different groups. For a candidate change-point t, we define:

$$R_G(t) = \sum_{(i,j)\in G} I_{g_i(t)\neq g_j(t)},$$

where $g_i(t)$ is a group indicator:

$$g_i(t) = \begin{cases} 0, & i \leq t, \\ 1, & i > t. \end{cases}$$

The indicator function $I_{g_i(t) \neq g_j(t)}$ equals 1 if the edge (i, j) connects nodes in different groups. For the changed interval hypothesis, the statistic is extended as:

$$R_G(t_1, t_2) = \sum_{(i,j) \in G} I_{g_i(t_1,t_2) \neq g_j(t_1,t_2)},$$

where $g_i(t_1, t_2) = I_{t_1 < i \leq t_2}$.

2.2.4 Standardization and Scan Statistic

To compare $R_G(t)$ across different t, we standardize it:

$$Z_G(t) = -\frac{R_G(t) - \mathbb{E}[R_G(t)]}{\sqrt{\mathrm{Var}(R_G(t))}}.$$

Here, $\mathbb{E}[R_G(t)]$ and $\mathrm{Var}(R_G(t))$ are the expectation and variance of $R_G(t)$ under the null hypothesis H_0. Specifically, under the null hypothesis H_0, the joint distribution of $\{y_i : i = 1, \ldots, n\}$ is assumed to be invariant under permutation. The null distribution of $R_G(t)$ is approximated using a permutation distribution, which assigns equal probability to each of the $n!$ permutations of $\{y_i : i = 1, \ldots, n\}$. $\mathbb{E}[R_G(t)]$ and $\mathrm{Var}(R_G(t))$ are the expectation and variance of $R_G(t)$ have explicit expressions under the null hypothesis H_0, defined as follows:

$$\mathbb{E}[R_G(t)] = p_1(t)|G|,$$

$$\mathrm{Var}(R_G(t)) = p_2(t)|G| + \left(\frac{1}{2}p_1(t) - p_2(t)\right) \sum_i |G_i|^2 + \left(p_2(t) - p_1^2(t)\right) |G|^2,$$

where

$$p_1(t) = \frac{2t(n-t)}{n(n-1)}, \quad p_2(t) = \frac{4t(n-t)(n-t-1)}{n(n-1)(n-2)(n-3)}.$$

Here, G_i denotes the set of edges connected to node x_i in G, and $|G_i|$ is the degree of node x_i. The scan statistic then maximizes the standardized statistic over all candidate change points to identify the most likely change point in the entire sequence:

$$\max_{n_0 \leq t \leq n_1} Z_G(t).$$

Finally,

$$\mathrm{P}(\max Z_G(t) > b),$$

is used to compute the p-value, where P represents the probability that the scan statistic exceeds b under the null hypothesis. This quantifies the statistical significance of the detected change points. The mathematical relationship between P and b is derived from the tail distribution of the scan statistic under the null hypothesis, as discussed in [CZ15]. For scenarios involving multiple change points, this graph-based scan can be extended using binary segmentation methods [Vos81; OVLW04]. This framework offers a robust and flexible method for detecting change-points in complex data structures. The choice of similarity metric and graph construction method plays a crucial role in the detection power. As it provides a general framework for change point detection, where the adjacency matrix of a weighted network is treated as high-dimensional data, we include this method as one of the competitors.

2.3 Kernel-Based Change-Point Detection

Kernel-based methods provide a flexible and powerful framework for conducting two-sample hypothesis testing in complex data by leveraging reproducing kernel Hilbert spaces (RKHS). These methods can effectively capture complex distributional changes, including location and scale changes. In this subsection, we introduce the theoretical foundation,

hypothesis testing framework, and detailed construction of kernel-based change-point detection statistics.

2.3.1 Kernel Function and RKHS Representation

The kernel function $k(x_i, x_j)$ measures the similarity between observations x_i and x_j. Common choices for kernel functions include the Gaussian kernel, $k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$, which emphasizes locality and is widely used for its smooth and continuous similarity measure; the linear kernel, $k(x_i, x_j) = x_i^\top x_j$, which captures direct linear relationships and is often used in linear models; and the polynomial kernel, $k(x_i, x_j) = (x_i^\top x_j + c)^p$, which introduces flexibility by mapping the data into higher-dimensional polynomial feature spaces, where c and p are constants. The kernel matrix K is constructed with elements $K_{ij} = k(x_i, x_j)$ for all i, j, encapsulating pairwise similarities in the data.

2.3.2 Test Statistic: Maximum Mean Discrepancy Based Method

In this section, we present kernel-based change-point detection algorithms (e.g., [HC07; LXDS15]) that utilize the Maximum Mean Discrepancy (MMD; [GBRSS06]) as the test statistic. Continuing from the notation established earlier, consider the independent observation sequence $\{x_i\}_{i=1}^n$. Our goal is to detect change points in the data distribution, which can be formalized through hypothesis testing. For a candidate change-point t, the observations are divided into two groups: $\{x_1, \ldots, x_t\}$ (Group 1) and $\{x_{t+1}, \ldots, x_n\}$ (Group 2). The test statistic is based on the MMD, which measures the difference between two distributions. The unbiased MMD statistic for t is given by:

$$\begin{aligned}\mathrm{MMD}_u^2(t) =& \frac{1}{t(t-1)} \sum_{i=1}^{t} \sum_{\substack{j=1 \\ j \neq i}}^{t} k(x_i, x_j) + \frac{1}{(n-t)(n-t-1)} \sum_{i=t+1}^{n} \sum_{\substack{j=t+1 \\ j \neq i}}^{n} k(x_i, x_j) \\ & - \frac{2}{t(n-t)} \sum_{i=1}^{t} \sum_{j=t+1}^{n} k(x_i, x_j).\end{aligned}$$

This statistic captures the internal similarity within each group and the dissimilarity between groups. To detect a change-point, the scan statistic maximizes $\mathrm{MMD}_u^2(t)$ over all possible t in the range $[n_0, n_1]$:

$$\max_{n_0 \le t \le n_1} \mathrm{MMD}_u^2(t).$$

2.3.3 Improved Test Statistic: GKCP

To address potential limitations of $\mathrm{MMD}_u^2(t)$, such as sensitivity to high-dimensional data and offset effects due to variance changes, the Generalized Kernel Change-Point (GKCP) statistic is introduced [SC24]. This statistic focuses on deviations in internal group similarities and is defined as:

$$\mathrm{GKCP}(t) = \begin{pmatrix} \alpha(t) - \mathbb{E}[\alpha(t)] \\ \beta(t) - \mathbb{E}[\beta(t)] \end{pmatrix}^\top \Sigma_t^{-1} \begin{pmatrix} \alpha(t) - \mathbb{E}[\alpha(t)] \\ \beta(t) - \mathbb{E}[\beta(t)] \end{pmatrix},$$

where $\alpha(t)$ and $\beta(t)$ are the mean kernel values within Group 1 and Group 2, respectively:

$$\alpha(t) = \frac{1}{t(t-1)} \sum_{i=1}^{t} \sum_{\substack{j=1 \\ j \neq i}}^{t} k(x_i, x_j), \quad \beta(t) = \frac{1}{(n-t)(n-t-1)} \sum_{i=t+1}^{n} \sum_{\substack{j=t+1 \\ j \neq i}}^{n} k(x_i, x_j).$$

The expectations $\mathbb{E}[\alpha(t)]$ and $\mathbb{E}[\beta(t)]$ are given by:

$$\mathbb{E}[\alpha(t)] = \mathbb{E}[\beta(t)] = \frac{1}{n(n-1)} R_0, \quad R_0 = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} k(x_i, x_j),$$

and $\Sigma_t = \mathrm{Var}([\alpha(t), \beta(t)])$ represents the covariance matrix of $\alpha(t)$ and $\beta(t)$. The scan statistic maximizes $\mathrm{GKCP}(t)$ over all possible t:

$$\max_{n_0 \le t \le n_1} \mathrm{GKCP}(t).$$

2.3.4 Thresholding and Hypothesis Testing

To test H_0 against H_1, the scan statistic is compared to a threshold derived from permutation tests. Specifically, if:

$$\max_{n_0 \leq t \leq n_1} \text{GKCP}(t) > \text{Threshold},$$

the null hypothesis H_0 is rejected, indicating the presence of a change-point at t. The detailed procedure for constructing the p-value can be found in [SC24].

The GKCP statistic overcomes the limitations of $\text{MMD}_u^2(t)$ by incorporating internal group deviations and normalizing them with respect to their expectations. It is robust to high-dimensional data and variance changes, enabling effective detection of both location and scale shifts. This robustness makes GKCP a powerful tool for identifying change-points in complex, high-dimensional data distributions. Given that the weighted network adjacency matrix can be treated as high-dimensional data, we incorporate this method as one of our competitors.

CHAPTER 3

Generative Models for Network Change-Point Detection

Generative models have become powerful tools for modeling complex data distributions, with widespread applications in various domains. Variational Autoencoders (VAEs, [Kin13]) offer a probabilistic framework that learns latent representations by maximizing a lower bound on data likelihood. Generative Adversarial Networks (GANs, [GPMXWOCB20]), on the other hand, employ a min-max game between a generator and a discriminator, enabling the generation of highly realistic samples. Diffusion models [HJA20] represent a newer class of generative approaches, which model data through iterative denoising processes, excelling in image and graph synthesis. Additionally, large language models (LLMs) like GPT [RWCLAS19] have demonstrated remarkable generative capabilities for sequential data and text, further showcasing the versatility of generative modeling techniques.

[KLLCP24] proposed a generative model for detecting change-points in binary networks, providing an effective framework for analyzing temporal changes in network structures. The model leverages a graph decoder based on a likelihood function suited for binary data and a neural network structure designed specifically for binary adjacency matrices.

Building on this foundation, this work extends the framework in [KLLCP24] to accommodate weighted networks and other graph types, including directed and undirected networks. The proposed extension introduces a generalized likelihood function that can handle different network data distributions, such as Poisson and Gaussian models for weighted edges. Additionally, we expand the neural network design in the graph decoder

to account for the structural diversity of various graph types. These modifications ensure the framework's applicability to a broader range of real-world scenarios while retaining the strengths of the original binary network model.

In the following, we describe the model specification and highlight how it generalizes the binary network approach in [KLLCP24].

3.1 Model Specification

Consider a node set $N = \{1, 2, \ldots, n\}$ and let $\boldsymbol{y} \in \mathbb{R}^{n \times n}$ represent an adjacency matrix for a graph. The adjacency matrix can encode binary or weighted relationships for all pairs $(i, j) \in \mathbb{Y} = N \times N$, with $\boldsymbol{y}_{ij}$ denoting the strength of the relation between nodes i and j. For undirected networks, we assume symmetry, i.e., $\boldsymbol{y}_{ij} = \boldsymbol{y}_{ji}$ for all $(i, j) \in \mathbb{Y}$.

Let $\boldsymbol{y}^t$ denote the network observed at a discrete time point t. The observed data consist of a sequence of networks $\{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^T\}$. For each network $\boldsymbol{y}^t$, we assume the existence of a latent variable $\boldsymbol{z}^t \in \mathbb{R}^d$ that generates the network via a graph decoder:

$$\boldsymbol{y}^t \sim P(\boldsymbol{y}^t | \boldsymbol{z}^t) = \prod_{(i,j) \in \mathbb{Y}} F(\boldsymbol{y}_{ij}^t; \boldsymbol{\theta}_{ij}(\boldsymbol{z}^t)),$$

where F represents the likelihood function, which can be specified as a Poisson, Bernoulli, or other distribution depending on the network type. The parameter $\boldsymbol{\theta}_{ij}(\boldsymbol{z}^t)$ governs the distribution of the dyad (i, j) and will be further detailed in Section 3.1.1. We assume that, conditioned on the latent variable $\boldsymbol{z}^t$, the network $\boldsymbol{y}^t$ is temporally independent.

The latent variable $\boldsymbol{z}^t$ is assumed to follow a learnable prior distribution:

$$\boldsymbol{z}^t \sim P(\boldsymbol{z}^t) = \mathcal{N}(\boldsymbol{z}^t; \boldsymbol{\mu}^t, \boldsymbol{I}_d),$$

where $\boldsymbol{\mu}^t \in \mathbb{R}^d$ and $\boldsymbol{I}_d$ is the identity matrix. Typically, the latent vector's dimension d is assumed to be significantly smaller than the number of nodes in the network. This ensures

that the latent variable $\boldsymbol{z}^t$ serves as a compact graph-level representation of the network $\boldsymbol{y}^t$, effectively capturing its structural characteristics in a lower-dimensional space.

3.1.1 Graph Decoder

The graph decoder $P(\boldsymbol{y}^t|\boldsymbol{z}^t)$ maps the latent variable $\boldsymbol{z}^t$ to the parameters of the likelihood function for each dyad. For instance, in the case of a Poisson-distributed network, the decoder parameterizes the Poisson rate $\boldsymbol{\theta}_{ij}(\boldsymbol{z}^t)$ as:

$$\boldsymbol{\theta}_{ij}(\boldsymbol{z}^t) = \boldsymbol{g}_{ij}\big(\boldsymbol{h}(\boldsymbol{z}^t)\big),$$

where $\boldsymbol{h} : \mathbb{R}^d \to \mathbb{R}^{n\times n}$ is a function implemented using neural networks. Multi-layer perceptrons (MLPs) with activation functions, such as the rectified linear unit (ReLU), are employed to introduce nonlinearity. With such design, the graph decoder $P(\boldsymbol{y}^t|\boldsymbol{z}^t)$ maps the latent variable $\boldsymbol{z}^t$ to the parameters of the likelihood function for each dyad.

The proposed framework generalizes the binary network model by accommodating different network types and distributions while preserving the core structure of the generative process. The flexibility in specifying the likelihood function F and parameterizing the decoder function $\boldsymbol{h}$ ensures applicability to a wide range of network analysis tasks. Figure 3.1 illustrates the overall workflow of the proposed framework, highlighting its adaptability to different scenarios.

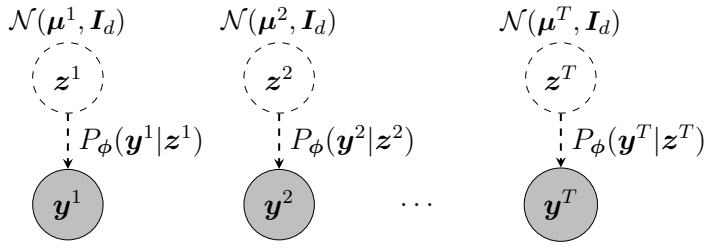

Figure 3.1: An overview of the proposed framework.

3.1.2 Binary Network Special Case

In [KLLCP24], the generative framework was instantiated specifically for binary networks. In this special case, the adjacency matrix entries $\boldsymbol{y}_{ij}^t$ are binary ($\boldsymbol{y}_{ij}^t \in \{0, 1\}$), representing the presence or absence of edges. The likelihood function F is specified as a Bernoulli distribution:

$$F(\boldsymbol{y}_{ij}^t; \boldsymbol{\theta}_{ij}(\boldsymbol{z}^t)) = \text{Bernoulli}(\boldsymbol{\theta}_{ij}(\boldsymbol{z}^t)),$$

where $\boldsymbol{\theta}_{ij}(\boldsymbol{z}^t)$ represents the probability of an edge existing between nodes i and j.

[KLLCP24] also parameterized $\boldsymbol{\theta}_{ij}(\boldsymbol{z}^t)$ using a special neural network structure to capture complex interactions between nodes with nodal embedding and dot product. Specifically $\boldsymbol{h}(\cdot)$ can be parameterized using latent matrix factorization, where the latent variable $\boldsymbol{z}^t$ is decomposed into matrices $\boldsymbol{U}^t \in \mathbb{R}^{n\times k}$ and $\boldsymbol{V}^t \in \mathbb{R}^{n\times k}$, allowing:

$$\boldsymbol{h}(\boldsymbol{z}^t) = \begin{cases} \boldsymbol{U}^t \boldsymbol{V}^{t\top}, & \text{for directed networks,} \\ \boldsymbol{U}^t \boldsymbol{U}^{t\top}, & \text{for undirected networks.} \end{cases}$$

This factorization ensures that the latent dimensions d and k are smaller than the number of nodes n, enabling a compact yet expressive representation of the network structure. Under this design, the graph-level latent variable $\boldsymbol{z}^t$ is transformed into node-level representations $\boldsymbol{U}^t$ and $\boldsymbol{V}^t$, which serve as intermediate steps before generating the network $\boldsymbol{y}^t$.

3.2 Change Points

Similar to the change-point definition in the CPD-STERGM framework, in the context of the generative model, we define the change points in terms of the prior parameters $\boldsymbol{\mu}^t \in \mathbb{R}^d$ for $t = 1, \ldots, T$. Specifically, let $\{C_k\}_{k=0}^{K+1} \subset \{1, 2, \ldots, T\}$ represent an ordered sequence of change points, where:

$$1 = C_0 < C_1 < \cdots < C_K < C_{K+1} = T,$$

such that:

$$\boldsymbol{\mu}^{C_k} = \boldsymbol{\mu}^{C_k+1} = \cdots = \boldsymbol{\mu}^{C_{k+1}-1}, \quad k = 0, \ldots, K,$$

$$\boldsymbol{\mu}^{C_k} \neq \boldsymbol{\mu}^{C_{k+1}}, \quad k = 0, \ldots, K-1, \quad \text{and} \quad \boldsymbol{\mu}^{C_{K+1}} = \boldsymbol{\mu}^{C_K}.$$

Our objective is to identify these unknown change points $\{C_k\}$ from the observed data.

Intuitively, we aim to characterize structural changes in the network through shifts in the prior distribution's parameters. These shifts in $\boldsymbol{\mu}^t$ capture transitions in the underlying graph structure over time, enabling us to detect and localize change points effectively.

3.3 Learning and Inference

We formulate the change-point detection problem as an optimization over a likelihood-based objective regularized by a Group-Fused Lasso penalty, which encourages sparsity in the differences of prior parameters between consecutive time points. Denoting the log-likelihood of the observed dynamic graphs $\boldsymbol{y}^1, \ldots, \boldsymbol{y}^T$ as $l(\boldsymbol{\phi}, \boldsymbol{\mu})$, the objective function is defined as:

$$\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}} =_{\boldsymbol{\phi},\boldsymbol{\mu}} -l(\boldsymbol{\phi}, \boldsymbol{\mu}) + \lambda \sum_{t=1}^{T-1} \|\boldsymbol{\mu}^{t+1} - \boldsymbol{\mu}^t\|_2,$$

where $\lambda > 0$ is a tuning parameter. The penalty promotes structural change detection by enforcing sparsity in the differences $\boldsymbol{\mu}^{t+1} - \boldsymbol{\mu}^t$, while the ℓ_2-norm allows simultaneous changes across multiple dimensions of the latent variable. This grouping effect is crucial for capturing complex changes in dynamic networks.

To solve the optimization problem, we adopt an Alternating Direction Method of Multipliers (ADMM, [BPCPE11]) framework. By introducing a slack variable $\boldsymbol{\nu}$, the objective is reformulated as a constrained optimization problem:

$$\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}} =_{\boldsymbol{\phi},\boldsymbol{\mu}} -l(\boldsymbol{\phi}, \boldsymbol{\mu}) + \lambda \sum_{t=1}^{T-1} \|\boldsymbol{\nu}^{t+1} - \boldsymbol{\nu}^t\|_2 \quad \text{subject to } \boldsymbol{\mu} = \boldsymbol{\nu}.$$

The augmented Lagrangian is defined, and updates for $\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\nu}$, and the dual variables are

derived. Details of these updates, including how the slack variable is incorporated and parameterized, are provided in [KLLCP24]. Here we illustrate the updates of the graph decoder parameter $\boldsymbol{\phi}$, which is able to be performed via back-propagation. The gradient of the objective with respect to $\boldsymbol{\phi}$ is given by:

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}) = -\sum_{t=1}^{T} \mathbb{E}_{P(\boldsymbol{z}^t|\boldsymbol{y}^t)} \Big(\nabla_{\boldsymbol{\phi}} \log P_{\boldsymbol{\phi}}(\boldsymbol{y}^t|\boldsymbol{z}^t) \Big).$$

Using the samples $\{\boldsymbol{z}_u^t\}_{u=1}^s$ from the posterior $P(\boldsymbol{z}^t|\boldsymbol{y}^t)$, the gradient can be approximated as:

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}) \approx -\sum_{t=1}^{T} \frac{1}{s} \sum_{u=1}^{s} \nabla_{\boldsymbol{\phi}} \log P_{\boldsymbol{\phi}}(\boldsymbol{y}^t|\boldsymbol{z}_u^t).$$

This allows the graph decoder $P_{\boldsymbol{\phi}}(\boldsymbol{y}^t|\boldsymbol{z}^t)$ to be updated iteratively using standard gradient-based optimization techniques. However, the updates for $\boldsymbol{\phi}$ include computing expectations under the posterior distribution $P(\boldsymbol{z}^t|\boldsymbol{y}^t) \propto P(\boldsymbol{y}^t|\boldsymbol{z}^t) \times P(\boldsymbol{z}^t)$. To approximate these expectations efficiently, we employ Langevin Dynamics sampling [PHNZW20]:

$$\boldsymbol{z}_{\tau+1}^t = \boldsymbol{z}_\tau^t + \delta \big[\nabla_{\boldsymbol{z}^t} \log P_{\boldsymbol{\phi}}(\boldsymbol{y}^t|\boldsymbol{z}^t) - (\boldsymbol{z}_\tau^t - \boldsymbol{\mu}^t) \big] + \sqrt{2\delta}\boldsymbol{\epsilon},$$

where $\delta > 0$ is the step size, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ introduces stochasticity for exploration. This approach approximates the likelihood while ensuring computational tractability.

In summary, the learning procedure alternates between optimizing the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\mu}$ and updating the slack variables using ADMM, with posterior expectations approximated through Langevin Dynamics. This iterative process continues until convergence, ensuring robust change-point detection. Notably, the log-likelihood can be easily adapted to different choices of likelihood functions given the latent vector $\boldsymbol{z}$, making various likelihood formulations as a flexible plug-in component for our proposed framework.

3.4 Model Selection and Change Point Localization

3.4.1 Model Selection

For model selection, we use Cross-Validation to tune λ. The time series of graphs is split into training (odd-indexed time points) and testing (even-indexed time points) sets. For each candidate λ, model parameters are learned on the training set, and the log-likelihood on the testing set is evaluated as:

$$\sum_{t=1}^{T} \log P(\boldsymbol{y}^t) \approx \sum_{t=1}^{T} \log \Big[\frac{1}{s} \sum_{u=1}^{s} \prod_{(i,j)\in\mathbb{Y}} P_{\boldsymbol{\phi}}(\boldsymbol{y}_{ij}^t | \boldsymbol{z}_u^t) \Big],$$

where Monte Carlo samples $\{\boldsymbol{z}_u^t\}_{u=1}^s$ are drawn from the prior distribution $P(\boldsymbol{z}^t)$. The λ maximizing the testing log-likelihood is selected. Finally, the parameters are re-learned using the full dataset with this λ, yielding the final set of detected change points.

3.4.2 Change Point Localization

Once parameters are learned, change points are localized by examining shifts in the prior parameters $\hat{\boldsymbol{\mu}}$. The differences between consecutive time points are calculated as:

$$\Delta\hat{\boldsymbol{\mu}}^t = \|\hat{\boldsymbol{\mu}}^t - \hat{\boldsymbol{\mu}}^{t-1}\|_2, \quad t = 2, \ldots, T.$$

These differences are standardized:

$$\Delta\hat{\boldsymbol{\zeta}}^t = \frac{\Delta\hat{\boldsymbol{\mu}}^t - \text{median}(\Delta\hat{\boldsymbol{\mu}})}{\text{std}(\Delta\hat{\boldsymbol{\mu}})},$$

and a data-driven threshold is defined:

$$\mathcal{T}_{\text{thr}} = \text{mean}(\Delta\hat{\boldsymbol{\zeta}}) + \mathcal{Z}_q \times \text{sd}(\Delta\hat{\boldsymbol{\zeta}}),$$

where $\mathcal{Z}_q$ is the $q\%$ quantile of the standard normal distribution. A time point C_k is identified as a change point if:

$$\Delta\hat{\boldsymbol{\zeta}}^{C_k} > \mathcal{T}_{\text{thr}}.$$

This approach ensures that significant structural changes are detected, while standardized differences close to zero are ignored.

3.5 Simulated Data Experiments

This section evaluates the performance of the proposed weighted graph change point detection method through comprehensive experiments. Given the limited availability of algorithms specifically designed for weighted graph change point detection, we compare our method with CPD-STERGM, a state-of-the-art approach for change point detection in binary graphs. Additionally, we include two widely used general-purpose change point detection methods: graph-based change point detection (gSeg) and kernel-based change point detection (kerSeg).

We evaluate our model's performance on weighted data derived from simulated counted networks. Following the experimental design in [KLLCP24], we simulate dynamic graph time series using two distinct models: the Stochastic Block Model (SBM, [HLL83]), and a Recurrent Neural Network (RNN, [Elm90]). These models represent sparse and dense graph time series, providing a comprehensive evaluation of the methods under different scenarios. Specifically, we generate dynamic graphs of size $n = 50$ nodes over a time span of $T = 100$. In each simulation, the generation mechanism changes at three predetermined change points located at $t = 26$, $t = 51$, and $t = 76$. The corresponding intervals are $[1, 25]$, $[26, 50]$, $[51, 75]$, and $[76, 100]$, creating a total of $K = 3$ change points. Each experiment is replicated across 30 Monte Carlo trials, and we report the mean and standard deviation of the evaluation metrics across these trials. Our design specifically constrains most parameters of the Poisson distribution to lie within the range $[0, 1]$, ensuring that the weighted scenario can be effectively transferred to binary graphs through hard thresholding,

where all edges with weights greater than 0 are set to 1, and edges with weights equal to 0 remain 0. This approach is designed to maintain fairness in the comparison.

We use three metrics to evaluate the performance of each algorithm. The first metric evaluates the discrepancy between the number of true and detected change points by calculating the absolute difference $|\hat{K} - K|$, where $\hat{K}$ represents the number of change points identified by the algorithm, and K is the true number of change points. This metric provides a direct measure of how well the method estimates the total number of change points, with smaller values indicating better performance.

The second and third metrics quantify the temporal accuracy of the detected change points using a modified Hausdorff distance, which has been employed in prior works such as [MYWR21; GA18; BKLMW09]. The one-sided Hausdorff distance measures the maximum temporal distance from any true change point to its nearest detected change point:

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

where $\mathcal{C}$ and $\hat{\mathcal{C}}$ denote the sets of true and detected change points, respectively. This metric evaluates the extent to which the algorithm can accurately localize the true change points. Additionally, we report the reversed one-sided Hausdorff distance:

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

This counterpart assesses the maximum distance from any detected change point to its nearest true change point. If no change points are detected (i.e., $\hat{\mathcal{C}} = \emptyset$), the convention is to assign $d(\hat{\mathcal{C}} \mid \mathcal{C}) = \infty$ and $d(\mathcal{C} \mid \hat{\mathcal{C}}) = -\infty$, indicating complete failure to detect any change points.

The third metric evaluates the overlap between the true and detected time partitions. Following the definition in [VW20], this metric calculates the degree to which the intervals formed by consecutive change points in the detected partition align with those in the true

partition:

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

where $\mathcal{G}$ and $\mathcal{G}'$ represent the true and detected partitions, respectively, and $\mathcal{A}$, $\mathcal{A}'$ are individual intervals within the partitions. This metric measures how well the detected time intervals cover the true intervals, with higher values indicating better coverage and alignment.

The results, including the mean and standard deviation of these metrics across the Monte Carlo trials, are reported in the following sections. Our proposed method, denoted Weighted-CPDLatent, employs a data-driven thresholding approach based on the 90th percentile of the standard normal distribution. The latent dimensions for the graph decoder are set to $d = 10$ and $k = 5$, balancing computational complexity and detection accuracy. The likelihood function is modeled using a Poisson distribution.

Scenario 1: Poisson SBM

Poisson parameter matrices $P, Q \in \mathbb{R}^{n \times n}$ are constructed and they are defined as

$$P_{ij} = \begin{cases} 0.5, & i, j \in \mathcal{B}_l, l \in [1, 2, 3], \\ 4, & \text{otherwise}, \end{cases} \quad \text{and} \quad Q_{ij} = \begin{cases} 0.75, & i, j \in \mathcal{B}_l, l \in [1, 2, 3], \\ 3.4, & \text{otherwise}, \end{cases}$$

where $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ are three evenly sized clusters that form a partition of $\{1, \ldots, n\}$. Then a sequence of matrices $E^t \in [0, 1]^{n \times n}$ are arranged for $t = 1, \ldots, T$ such that

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

Lastly, the networks are generated with $\rho = 0.5$, representing a strong temporal dependency, which aligns with patterns commonly observed in real-world temporal data. For $t =$

$1, \ldots, T-1$, we let $y_{ij}^t \sim \text{Poisson}(E_{ij}^t)$ and

$$y_{ij}^{t+1} \sim \text{Poisson}\left(\rho E_{ij}^t + (1-\rho) y_{ij}^t\right).$$

Figures 3.2 and 3.3 offer description for the generated graphs. Our simulated graphs are challenging to differentiate through trivial observation, underscoring the non-triviality of our change-point detection simulation setting. Experimental results in Table 3.1 demonstrate that both the CPD-STERGM algorithm and our proposed algorithm achieve nearly perfect detection accuracy, with our method slightly outperforming in terms of Hausdorff distance, ranking as the top-performing algorithm. In contrast, the performance of gSeg and kerSeg in this setting is poor, as they consistently overestimate the number of potential change points across the sequence.

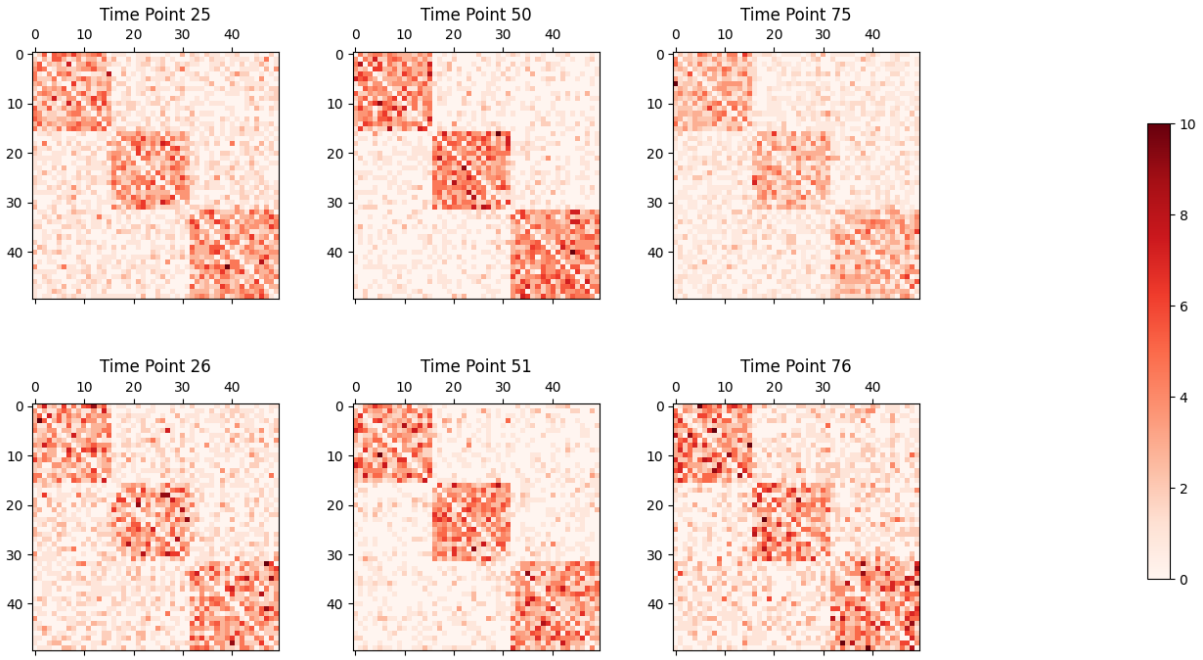

Figure 3.2: Examples of graphs generated using SBM with $n = 50$. The first row shows graphs at $t = 25$, $t = 50$, and $t = 75$, arranged from left to right. The second row displays graphs at $t = 26$, $t = 51$, and $t = 76$, representing the change points.

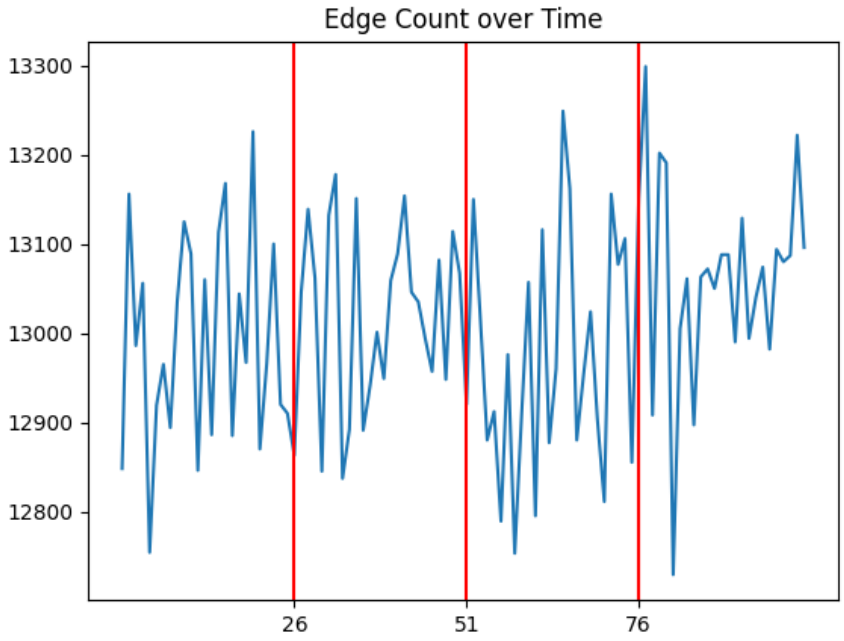

Figure 3.3: Total edge weights over time for graphs generated using SBM. The plot depicts the sum of all edge weights at each time point. The red vertical lines represent the detected change points, located at $t = 26$, $t = 51$, and $t = 76$.

Table 3.1: Performance comparison of methods for change point detection on dynamic networks simulated from SBM. Results are averaged over 30 Monte Carlo simulations and reported as means (standard deviations) across performance metrics.

Method	$\lvert\hat{K} - K\rvert \downarrow$	$d(\hat{C}\lvert C) \downarrow$	$d(C\lvert\hat{C}) \downarrow$	$\mathcal{C}(G, G') \uparrow$
CPD-STERGM$_{p=4}$	0.0 (0.0)	1.0 (0.0)	1.0 (0.0)	96.5%
CPD-STERGM$_{p=6}$	0.1 (0.3)	1.0 (0.0)	1.8 (2.4)	97.1%
gSeg	12.2 (0.3)	0.3 (0.5)	19.0 (0.0)	27.8%
kerSeg	6.8 (0.8)	0.0 (0.2)	16.6 (1.8)	44.7%
Weighted-CPDLatent	0.3 (0.6)	0.0 (0.0)	3.5 (7.2)	**97.3%**

Scenario 2: RNN

In this scenario, we employ Recurrent Neural Networks (RNNs) to generate sequences of weighted dynamic networks. Specifically, latent variables are sampled from predefined priors, and the RNN is initialized with uniform weights. The network graphs are subsequently generated through matrix multiplication using the RNN outputs. The parameters for the predefined priors are as follows:

$$z^t \sim \begin{cases} \mathcal{N}(-1, 0.1I_d), & \text{if } t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ \mathcal{N}(5, 0.1I_d), & \text{if } t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

At each time point, the neural network's output serves as the parameter for an edge-wise Poisson distribution, which is used to generate weighted edges. Similar to the previous scenario, the RNN-based simulation enforces a time-dependent mechanism across dynamic networks. Figures 3.4 and 3.5 offer description of generated networks under this setting.

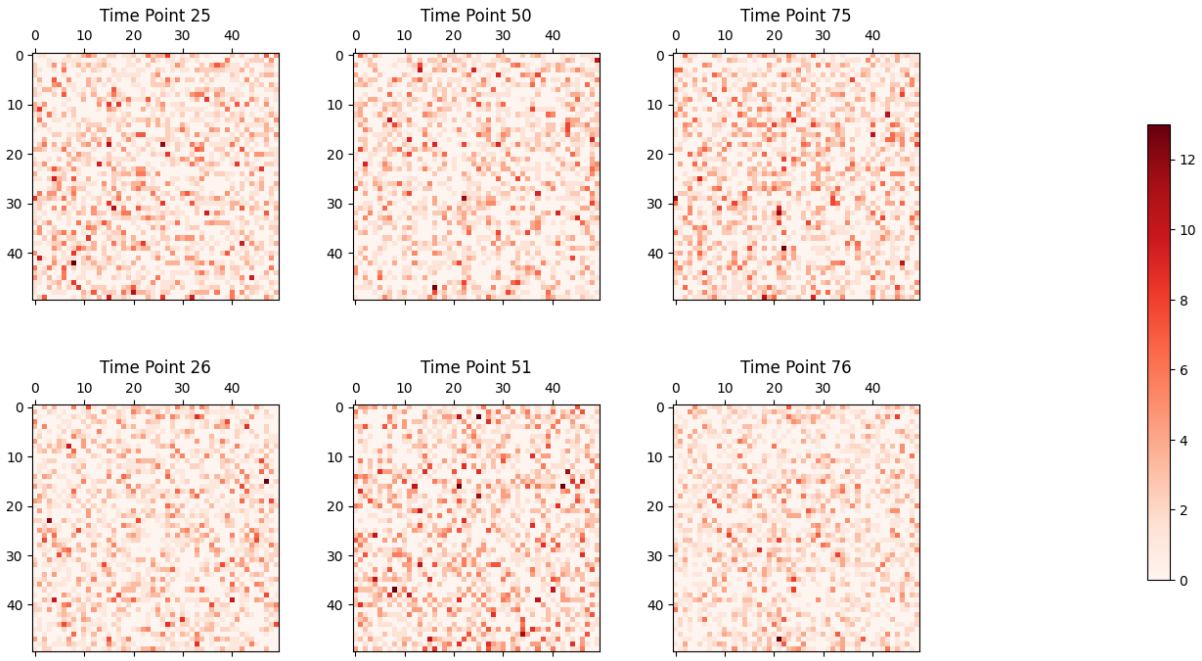

Figure 3.4: Examples of graphs generated using RNN with $n = 50$. The first row shows graphs at $t = 25$, $t = 50$, and $t = 75$, arranged from left to right. The second row displays graphs at $t = 26$, $t = 51$, and $t = 76$, representing the change points.

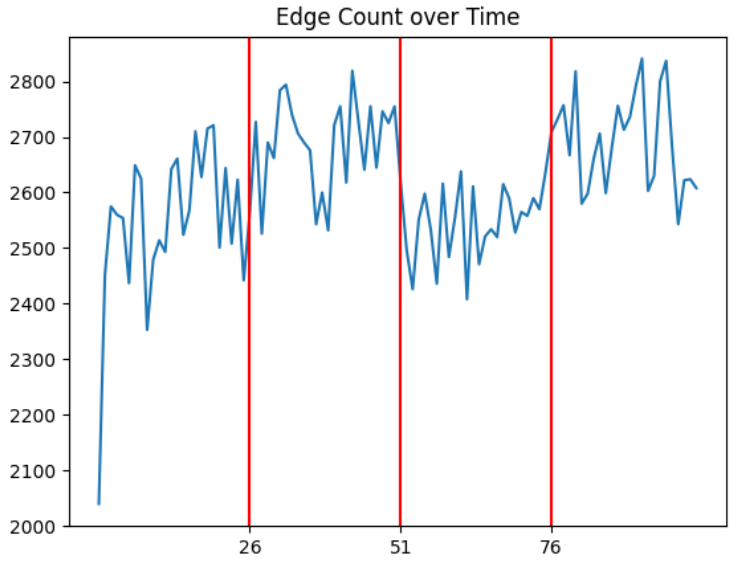

Figure 3.5: Total edge weights over time for graphs generated using RNN. The plot depicts the sum of all edge weights at each time point. The red vertical lines represent the detected change points, located at $t = 26$, $t = 51$, and $t = 76$.

Table 3.2: Performance comparison of methods for change point detection on dynamic networks simulated from RNN. Results are averaged over 30 Monte Carlo simulations and reported as means (standard deviations) across performance metrics.

Method	$\lvert\hat{K} - K\rvert \downarrow$	$d(\hat{C}\lvert C) \downarrow$	$d(C\lvert\hat{C}) \downarrow$	$\mathcal{C}(G, G') \uparrow$
CPD-STERGM$_{p=4}$	2.1 (1.0)	6.7 (6.1)	11.5 (4.8)	68.8%
CPD-STERGM$_{p=6}$	1.2 (0.8)	15.4 (10.2)	12.9 (3.2)	64.3%
gSeg	3.0 (0.0)	Inf (NA)	Inf (NA)	0%
kerSeg	0.9 (0.6)	1.8 (0.4)	5.1 (3.3)	90.1%
Weighted-CPDLatent	0.1 (0.2)	1.6 (0.6)	2.3 (2.5)	**94.9%**

The performance of the models, as shown in Table 3.2, indicates that under the RNN-generated graph setting, our proposed method achieves the best performance compared to competitors. Specifically, the performance of STERGM-based change-point detection methods shows a significant decline compared to the previous experimental setting, while gSeg consistently fails to identify even a single potential change point across all Monte Carlo simulations using binary search. Notably, the kernel-based change-point algorithm performs relatively well in this setting, achieving more than 90% overlap of the time partitions between the detected sequences and the truth.

In conclusion, our proposed method demonstrates state-of-the-art performance for change-point detection tasks in Poisson-distributed weighted graph settings.

3.6 Real Data Experiments

We evaluate our model using the Enron email dataset [Uni04], a well-known real-world example often used in network analysis [PCMP05; PC15; PC15; KLLCP24]. This dataset records detailed timestamps of email communications among employees during the period of Enron's bankruptcy crisis. This scenario highlights the importance of incorporating weighted networks, where the strength of connections between employees is represented by the frequency of their email exchanges. The weights are calculated by aggregating the number of emails exchanged over specified time intervals.

To maintain comparability with the previous work in [KLLCP24], we follow a similar network construction methodology. Specifically, we focus on the top 100 employees with the highest email communication frequencies. Weekly email exchanges between each pair of employees are aggregated to construct a temporal graph sequence spanning 100 weeks. Each graph is represented by a 100×100 weighted adjacency matrix. Using the same model structure and hyperparameters as described in the simulation experiments, we perform change point detection. The results of the experiment are presented in Figure 3.6.

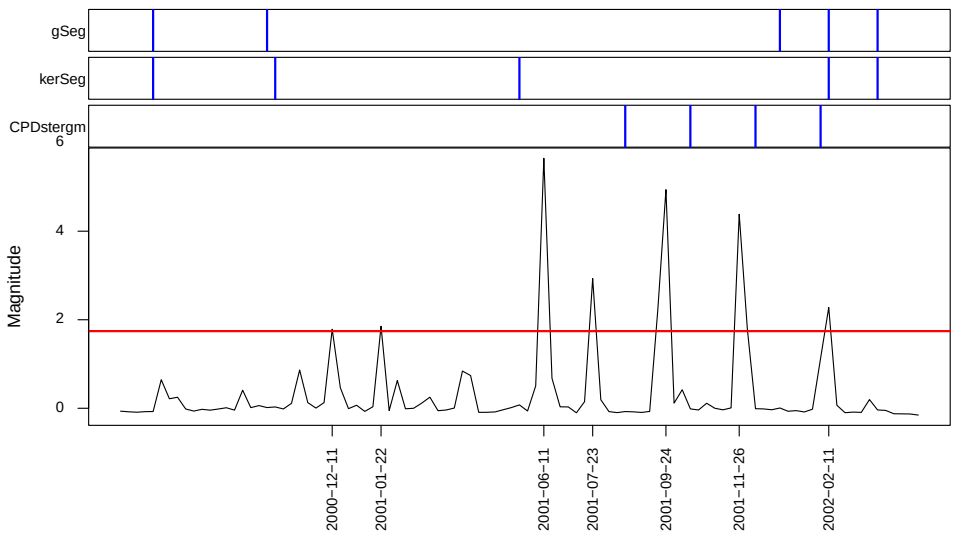

Figure 3.6: Detected change points on the Enron Email Data. The red line represents the data-driven threshold calculated at the 90% quantile, while the change-point detection results from competitor algorithms are shown in the upper panels as blue lines.

We list the major events potentially corresponding to the detected change points in Table 3.3. As shown, our proposed algorithm achieves superior change-point detection by identifying points that more closely align with real-world events, compared to results from other competitors. Specifically, gSeg and kerSeg incorrectly estimate the change point too early, in July 2000, when the fall of Enron had not yet begun, while CPD-STERGM provides detection results that occur later than the significant event's actual timing. This underscores the importance of incorporating weighted networks in change point detection, as they capture more informative patterns and relationships, in contrast to relying solely on binary networks.

Table 3.3: Detected change points (CP) and their corresponding potential nearby events.

Detected CP	**Potential Nearby Events**
2000-12-11	2000-12-13 CEO transition
2001-01-22	2001-01-22 Quarterly analyst conference
2001-06-11	2001-06-21 CEO publicly confronted
2001-07-23	2001-07-24 Analyst meeting
2001-09-24	2001-09-26 Employee Meeting
2001-11-26	2001-11-29 Stock price turmoil & Dynegy withdraws acquisition
2002-02-11	2001-02-07 Court Hearing Period

3.7 Discussion

This work focuses on change point detection for temporal weighted graphs using generative models. We extend the binary network structure proposed in [KLLCP24] by introducing a general decoder-only neural network framework for modeling weighted graphs. Using Poisson distribution as an example, both simulation experiments and real-world applications demonstrate the effectiveness of the proposed method.

A potential limitation of this approach lies in the requirement that the likelihood distribution governing the generation of edges in the graph, given the corresponding latent vector, should closely approximate the true data generation process. In real-world applications, this assumption may be challenging to satisfy. As a future direction, it is important to evaluate the robustness of the method when there is a significant mismatch between the assumed likelihood distribution and the true edge-generation distribution. Additionally, approaches that do not rely on specific designs for the likelihood distribution should be considered. Finally, as the proposed method depends on neural networks, exploring more complex network architectures could further enhance the performance of the model.

REFERENCES

[BBM20] Monika Bhattacharjee, Moulinath Banerjee, and George Michailidis. "Change point estimation in a dynamic stochastic block model". In: *Journal of machine learning research* 21.107 (2020), pp. 1–59.

[BF10] José M Bioucas-Dias and Mário AT Figueiredo. "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing". In: *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*. IEEE. 2010, pp. 1–4.

[BKLMW09] Leif Boysen, Angela Kempe, Volkmar Liebscher, Axel Munk, and Olaf Wittich. "Consistencies and rates of convergence of jump-penalized least squares estimators". In: (2009).

[BPCPE11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine learning* 3.1 (2011), pp. 1–122.

[CZ15] Hao Chen and Nancy Zhang. "Graph-based change-point detection". In: (2015).

[Elm90] Jeffrey L Elman. "Finding structure in time". In: *Cognitive science* 14.2 (1990), pp. 179–211.

[FCG10] Pedro A Forero, Alfonso Cano, and Georgios B Giannakis. "Consensus-based distributed linear support vector machines". In: *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*. 2010, pp. 35–46.

[FR79] Jerome H Friedman and Lawrence C Rafsky. "Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests". In: *The Annals of Statistics* (1979), pp. 697–717.

[GA18] Damien Garreau and Sylvain Arlot. "Consistent change-point detection with kernels". In: (2018).

[GBRSS06] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. "A kernel method for the two-sample-problem". In: *Advances in neural information processing systems* 19 (2006).

[GPMXWOCB20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[HC07] Zaid Harchaoui and Olivier Cappé. "Retrospective mutiple change-point estimation with kernels". In: *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*. IEEE. 2007, pp. 768–772.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[HLL83] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. "Stochastic blockmodels: First steps". In: *Social networks* 5.2 (1983), pp. 109–137.

[KH14] Pavel N Krivitsky and Mark S Handcock. "A separable model for dynamic networks". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 76.1 (2014), pp. 29–46.

[Kin13] Diederik P Kingma. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[KLCP23] Yik Lun Kei, Hangjian Li, Yanzhen Chen, and Oscar Hernan Madrid Padilla. "Change point detection on a separable model for dynamic networks". In: *arXiv preprint arXiv:2303.17642* (2023).

[KLLCP24] Yik Lun Kei, Jialiang Li, Hangjian Li, Yanzhen Chen, and Oscar Hernan Madrid Padilla. "Change Point Detection in Dynamic Graphs with Generative Model". In: *arXiv preprint arXiv:2404.04719* (2024).

[LBFM21] Federico Larroca, Paola Bermolen, Marcelo Fiori, and Gonzalo Mateos. "Change point detection in weighted and directed random Dot Product Graphs". In: *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE. 2021, pp. 1810–1814.

[LKR13] Dean Lusher, Johan Koskinen, and Garry Robins. *Exponential random graph models for social networks: Theory, methods, and applications*. Cambridge University Press, 2013.

[LXDS15] Shuang Li, Yao Xie, Hanjun Dai, and Le Song. "M-statistic for kernel change-point detection". In: *Advances in Neural Information Processing Systems* 28 (2015).

[MYWR21] Oscar Hernan Madrid Padilla, Yi Yu, Daren Wang, and Alessandro Rinaldo. "Optimal nonparametric change point analysis". In: (2021).

[Nic08] Christine Leigh Myers Nickel. "Random dot product graphs a model for social networks". PhD thesis. Johns Hopkins University, 2008.

[OP09] Tore Opsahl and Pietro Panzarasa. "Clustering in weighted networks". In: *Social networks* 31.2 (2009), pp. 155–163.

[OVLW04] Adam B Olshen, E Seshan Venkatraman, Robert Lucito, and Michael Wigler. "Circular binary segmentation for the analysis of array-based DNA copy number data". In: *Biostatistics* 5.4 (2004), pp. 557–572.

[PC15] Leto Peel and Aaron Clauset. "Detecting change points in the large-scale structure of evolving networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1. 2015.

[PCMP05] Carey E Priebe, John M Conroy, David J Marchette, and Youngser Park. "Scan statistics on enron graphs". In: *Computational & Mathematical Organization Theory* 11 (2005), pp. 229–247.

[PHNZW20] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. "Learning latent space energy-based prior model". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21994–22008.

[Ros05] Paul R Rosenbaum. "An exact distribution-free test comparing two multivariate distributions based on adjacency". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.4 (2005), pp. 515–530.

[RWCLAS19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[SC24] Hoseung Song and Hao Chen. "Practical and powerful kernel-based change-point detection". In: *IEEE Transactions on Signal Processing* (2024).

[TSRZK05] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. "Sparsity and smoothness via the fused lasso". In: *Jour-*

nal of the Royal Statistical Society Series B: Statistical Methodology 67.1 (2005), pp. 91–108.

[Uni04] Carnegie Mellon University. *Enron Email Dataset*. https://www.cs.cmu.edu/~enron/. Accessed: 2024-12-07. 2004.

[Vos81] Lyudmila Yur'evna Vostrikova. "Detecting "disorder" in multidimensional random processes". In: *Doklady akademii nauk*. Vol. 259. 2. Russian Academy of Sciences. 1981, pp. 270–274.

[VW20] Gerrit JJ Van den Burg and Christopher KI Williams. "An evaluation of change point detection algorithms". In: *arXiv preprint arXiv:2003.06222* (2020).

[YS07] Stephen J Young and Edward R Scheinerman. "Random dot product graph models for social networks". In: *International Workshop on Algorithms and Models for the Web-Graph*. Springer. 2007, pp. 138–149.