Diploma Project

DYNAMIC OVERLAPPING COMMUNITY DETECTION ON SPARSE NETWORKS

Antreas Laos

University of Cyprus
Department of Computer Science

Department of Computer Science

May 2026

UNIVERSITY OF CYPRUS

Faculty of Pure and Applied Sciences

Department of Computer Science

DYNAMIC OVERLAPPING COMMUNITY DETECTION ON SPARSE NETWORKS

Antreas Laos

Supervisors:
Dr. Xenia Miscouridou, Dr. Chryssis Georgiou

The Thesis was submitted in partial fulfillment of the requirements for obtaining the Computer Science degree of the Department of Computer Science of the University of Cyprus

May 2026

I declare that this dissertation and any accompanying software are my own original work, conducted in full compliance with the University of Cyprus Code of Conduct for Research (https://www.ucy.ac.cy/legislation/kodikas-deontologias-kai-politikes/), and with full accountability on my part for the accuracy, integrity, and validity of all content

Acknowledgements

Working on this thesis has been an intellectually stimulating and deeply rewarding journey. The process has not only broadened my understanding of the field but has also shaped me as a researcher in ways I did not anticipate at the outset.

Firstly, I would like to thank my advisors, Xenia Miscouridou and Chryssis Georgiou. Xenia, for supporting me throughout this process, for the countless hours that she invested in my development, and for the formative guidance throughout the years that shaped my drive for research. Chryssis, for his thoughtful advice and direction, which helped mould this thesis into what it has become, and for consistently bringing a fresh perspective to our discussions.

I would also like to extend my thanks to Francesca Panero for her valuable contributions to this work, and for lending an international dimension to what might otherwise have remained a local endeavor.

Finally, and most importantly, I would like to thank my family for their unconditional support and for instilling in me the discipline and work ethic that made the completion of this thesis possible.

The dynSNetOC model, together with additional theoretical results on its asymptotic sparsity and degree behavior not included here, has been written up in Miscouridou, Panero & Laos (2025), ”Dynamic sparse graphs with overlapping communities”, arXiv:2512.10717.

ABSTRACT

Real-world networks are typically sparse, exhibit heavy-tailed degree distributions, contain overlapping community structure, and evolve over time. No existing model jointly captures all four of these properties within a single framework: probabilistic constructions handle sparsity and degree heterogeneity but rarely overlap and dynamics together, while graph neural networks scale to large data but have not been extended to overlapping community detection in the dynamic setting. This thesis closes the gap from both methodological directions. The first contribution, dynSNetOC, is a Bayesian nonparametric model for dynamic sparse graphs with overlapping communities, built from vectors of completely random measures coupled through a latent Markov process and fit with the No-U-Turn Sampler. The second contribution, dNOCD, is a temporal extension of the NOCD model in which community affiliations are temporally coupled and trained end-to-end against a Bernoulli-Poisson likelihood. Evaluated on a Reuters Terror word co-occurrence network and a European commercial flights network, both models recover interpretable, temporally consistent overlapping communities where dynamic stochastic blockmodel baselines fail, while exposing a clear trade-off: the Bayesian model delivers calibrated uncertainty and faithful power-law degree behaviour at the cost of computational time, whereas the neural model scales and admits node features but recovers neither uncertainty nor the degree distribution.

Keywords: dynamic community detection; overlapping communities; sparse networks; Bayesian nonparametrics; graph neural networks

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

TABLE OF CONTENTS

1.1 Overview

Networks or graphs (used interchangeably throughout the thesis) are mathematical structures that encode relationships between a set of entities, and they appear across nearly every quantitative domain of modern science. A social network records friendships or interactions between people; a citation network captures the flow of ideas between scientific papers; a biological network describes interactions between proteins or genes; a transportation network represents the routes connecting cities or airports; an economic network describes the flows of capital between individuals or firms. The widespread availability of data of this kind, together with the sheer size of the networks involved, has driven sustained interest in statistical methods that can extract meaningful structure from them [1].

Informally, a network consists of a set of nodes together with a set of edges connecting them. The degree of a node is the number of edges incident to it, and the rate at which the total number of edges grows with the number of nodes determines whether the graph is dense (edges scaling quadratically with nodes) or sparse (edges scaling subquadratically). Most real-world networks fall firmly in the sparse regime, and many exhibit heavy-tailed, often power-law, degree distributions: a small number of nodes accumulate a disproportionate share of the connections while the vast majority remain weakly connected [2].

A central task in statistical network modeling is community detection: the inference of latent groups of nodes that connect more densely within themselves than across the network. Communities reveal hidden organization. In social networks they correspond to friend groups or interest circles; in information networks to topic clusters; in biological networks to functional modules; in economic networks to tightly interacting groups of firms or markets. Two observations make this task substantially harder than the textbook formulation suggests. First, communities in real networks are typically overlapping rather than disjoint, for example a person belongs to family, work and hobby circles simultaneously, and a research paper sits at the intersection of several subfields. Second, real networks are rarely static [3]. New connections form and old ones dissolve; communities emerge, merge, drift, and disappear. A social tie that was central one year may be marginal the next, and a topic that dominated news coverage for a month may vanish entirely thereafter. Capturing this temporal dimension is essential whenever the goal is to understand how a system has evolved or to possibly predict where it is going.

Taken together, these observations point to four properties that a fully realistic model of dynamic community structure should exhibit: *sparse connectivity*, *power-law degree distributions*, *overlapping community affiliations*, and *time-varying node memberships and edges*.

1.2 Related Work

The existing literature on community detection can be organized, at a high level, along two methodological axes. The first is the probabilistic line of work, which builds explicit generative models of the graph and recovers community structure through posterior or likelihood-based inference. The stochastic blockmodel [4] and its overlapping [5] and dynamic [6, 7] counterparts are representative examples. More recently, the Caron–Fox family of models built from exchangeable random measures [8, 9] provides the first construction capable of generating sparse graphs with power-law degree distributions while preserving exchangeability. Models in this line are interpretable and support uncertainty quantification, but rely on Markov chain Monte Carlo for inference and scale poorly with network size. The second axis is the deep learning line of work, which parameterizes node affiliations with a graph neural network and trains them end-to-end by stochastic gradient descent. NOCD [10] is a representative static instance for overlapping community detection. Models in this line scale gracefully on GPU hardware and integrate naturally with node features, but yield point estimates rather than full posteriors and have not yet been extended to overlapping community detection in the dynamic setting. A more extensive exploration of the related work is given in Section 2.2.

1.3 Contributions

Neither paradigm currently offers a model that captures all four properties (sparsity, power-law degrees, overlap, and temporal evolution) within a single framework. This thesis closes that gap from both directions. The first contribution is dynSNetOC, a Bayesian nonparametric model for dynamic sparse graphs with overlapping communities, constructed from vectors of completely random measures coupled through a latent Markov process governing the evolution of node affiliations. The second contribution is dNOCD, a temporal extension of the NOCD graph neural network in which community affiliations are propagated across time through a graph encoder and temporal coupling and trained end-to-end against a Bernoulli–Poisson likelihood. The two models are developed independently, in Chapters 3 and 4 respectively, but evaluated on a matched pair of real-world datasets: a word co-occurrence network derived from the Reuters Terror corpus and a European commercial flights network. As such, the thesis tackles the problem of dynamic overlapping community detection on sparse graphs from both perspectives, providing a comparison of the two on the same datasets, highlighting the trade-offs between probabilistic and deep-learning approaches.

1.4 Outline

The remainder of this thesis is organized as follows.

Chapter 2 reviews the technical background and related work for both methodological paradigms, covering the relevant probabilistic models of network structure on one side and graph neural networks for community detection on the other.

Chapters 3 and 4 address the same problem on the same data with different methodological tools, and the comparison between them is the central empirical contribution of the thesis. Chapter 3 develops dynSNetOC, deriving its generative construction from vectors of completely random measures, presenting the inference procedure based on the No-U-Turn Sampler, and reporting empirical results on synthetic data and on the Reuters Terror and European Flights datasets. Chapter 4 develops dNOCD, the temporal extension of NOCD, describing its graph convolutional architecture, its training objective, and its evaluation on the same two real-world datasets.

Chapter 5 concludes with a summary of the findings, a discussion of the trade-offs surfaced by the cross-paradigm comparison, and directions for future work.

2 Background and Related Work

TABLE OF CONTENTS

2.1 Background

This section covers the necessary technical background to understand the concepts described in the rest of the thesis. For a deeper dive into the topics described, we recommend the books Networks: An Introduction by Mark Newman [11] and Geometric Deep Learning by Bronstein et al. [12].

2.1.1 Graphs

We proceed with some typical definitions regarding graphs.

Definition 1 (Graph). *A graph (or network) is an ordered pair $\mathcal{G} = (V, E)$ where:*

- *V is a non-empty set whose elements are called nodes (or vertices);*
- *$E \subseteq V \times V$ is a set of edges, where each edge connects two nodes.*

A graph can also be represented by an adjacency matrix $A \in \{0,1\}^{|V|\times|V|}$, where $A_{ij} = 1$ if $(i, j) \in E$. $\mathcal{G}$ is considered an undirected graph if $A_{ij} = A_{ji}$ else it is considered directed.

Definition 2 (Multigraph). *A* multigraph *is a graph in which multiple edges between the same pair of nodes are allowed. It can be represented by an edge-count matrix* $N \in \mathbb{N}^{|V|\times|V|}$*, where* n_{ij} *denotes the number of edges from node* i *to node* j*. A standard (simple) graph is recovered by setting* $A_{ij} = \mathbf{1}[n_{ij} > 0]$.

Definition 3 (Sparsity). *The graph* G *is said to be* sparse *if the number of edges grows sub-quadratically with the number of nodes:*

$$|E| = o(|V|^2)$$

Definition 4 (Degree). *Let* $\mathcal{G} = (V, E)$ *be a graph. We define the degree of node* $i \in V$ *, denoted by* $deg(i)$*, as the number of edges incident to* i*:*

$$deg(i) = \sum_{j \in V} Aij$$

Definition 5 (Degree Distribution). *Let* p_k *be the fraction of nodes in a graph with degree* k*. We define the degree distribution as:*

$$P(k) = p_k = \frac{N_k}{N},$$

where N_k *is the number of nodes with degree* k *and* N *is the total number of nodes. Equivalently,* $P(k)$ *is the probability that a node chosen uniformly at random has degree* k*, and satisfies*

$$\sum_{k=0}^{\infty} P(k) = 1.$$

Definition 6 (Power-law Networks). *Power-law Networks also known as scale-free Networks, are graphs whose degree distribution follows a power law:*

$$P(k) \sim k^{-\gamma}$$

These types of networks are commonly found in real-world systems, where typically $2 < \gamma < 3$.

Definition 7 (Community). *Let* $\mathcal{G} = (V, E)$ *be a graph with* k *communities* $C = \{C_1, \ldots, C_k\}$*, where* C_i *is a subgraph of* $\mathcal{G}$ *where nodes in the same subgraph* C_i *are more densely connected* *with each other than with nodes in another subgraph C_j. If the communities are non-overlapping then $C_i \cap C_j = \emptyset\ \forall i \neq j$.*

Definition 8 (Overlapping Communities)**.** *The communities $\mathcal{C} = \{C_1, \ldots, C_k\}$ are said to be* overlapping *if there exist $i \neq j$ such that $C_i \cap C_j \neq \emptyset$, i.e. at least one node belongs to more than one community. Equivalently, each node $v \in V$ is associated with a non-negative affiliation vector $f_v \in \mathbb{R}^k_{\geq 0}$ whose entries quantify the strength of its membership to each community, and overlap corresponds to f_v having more than one non-zero entry. Overlapping community detection is the task of recovering these affiliation vectors from the observed graph.*

2.1.2 Bayesian Statistics

We provide some definitions around Bayesian Statistics that are relevant to the thesis.

Definition 9 (Markov Chain Monte Carlo (MCMC))**.** Markov Chain Monte Carlo *is a class of algorithms for sampling from a target probability distribution π by constructing a Markov chain $\{X_t\}_{t\geq 0}$ whose stationary distribution coincides with π. After a sufficient burn-in period, the samples X_t are approximately distributed according to π and can be used to estimate expectations via the ergodic average*

$$\mathbb{E}_\pi[f(X)] \approx \frac{1}{T}\sum_{t=1}^{T} f(X_t).$$

MCMC is especially useful in Bayesian inference, where the posterior is typically known only up to a normalising constant. Common instances include the Metropolis-Hastings algorithm, Gibbs sampling, and Hamiltonian Monte Carlo (in particular the No-U-Turn Sampler (NUTS) [13] used in this thesis).

Definition 10 (Hamiltonian Monte Carlo (HMC))**.** *Hamiltonian Monte Carlo is an MCMC method that uses gradient information of the target log-density to propose distant states with high acceptance probability, avoiding the diffusive random-walk behaviour of simpler samplers. HMC augments the target space with auxiliary* momentum *variables $\mathbf{p} \in \mathbb{R}^d$ and defines the* Hamiltonian

$$H(\boldsymbol{\theta}, \mathbf{p}) = \underbrace{-\log \pi(\boldsymbol{\theta})}_{U(\boldsymbol{\theta})\ \textit{(potential energy)}} + \underbrace{\tfrac{1}{2}\mathbf{p}^\top M^{-1}\mathbf{p}}_{K(\mathbf{p})\ \textit{(kinetic energy)}},$$

where M is a mass matrix (typically the identity or a diagonal preconditioning matrix). A proposal is generated by simulating Hamiltonian dynamics via the leapfrog integrator for L steps of size ε; the result is then accepted or rejected with a Metropolis-Hastings correction to account *for numerical discretisation error. Because the leapfrog trajectory follows the gradient of U, HMC can traverse the posterior in far fewer steps than gradient-free methods, making it well suited to high-dimensional, continuously-valued parameter spaces.*

Definition 11 (No-U-Turn Sampler (NUTS))**.** *The No-U-Turn Sampler [13] is an adaptive extension of HMC that eliminates the need to manually specify the number of leapfrog steps L. NUTS builds a binary tree of states forward and backward in simulated time, doubling its depth until the trajectory starts to double back on itself, i.e. until a* U-turn *criterion is detected:*

$$(\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-) \cdot \mathbf{p}^- < 0 \quad \textit{or} \quad (\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-) \cdot \mathbf{p}^+ < 0,$$

where $\boldsymbol{\theta}^+, \mathbf{p}^+$ and $\boldsymbol{\theta}^-, \mathbf{p}^-$ are the states at the forward and backward ends of the current tree. A candidate is drawn uniformly from the tree and accepted with a slice-based criterion that preserves the correct stationary distribution. NUTS also adapts the step size ε during a warm-up phase using a dual-averaging scheme, requiring no hand-tuning. In this thesis, NUTS is the inference algorithm used in Chapter 3 for the dynSNetOC model, as implemented in NumPyro.

Definition 12 (Exchangeability)**.** *A sequence of random variables $(X_1, X_2, \ldots)$ is* exchangeable *if its joint distribution is invariant under any finite permutation of the indices, i.e. for every n and every permutation σ of $\{1, \ldots, n\}$,*

$$(X_1, \ldots, X_n) \stackrel{d}{=} (X_{\sigma(1)}, \ldots, X_{\sigma(n)}).$$

In the context of graphs, an adjacency matrix A is jointly exchangeable *if $(A_{ij}) \stackrel{d}{=} (A_{\sigma(i)\sigma(j)})$ for every permutation σ of the node set, i.e. the distribution of the graph is invariant to relabelling of its nodes.*

Definition 13 (Poisson Point Process (PPP))**.** *A Poisson point process is a recipe for randomly scattering a set of points across some space Ω such that:*

(i) The number of points falling in any region $A \subseteq \Omega$ follows a Poisson distribution with mean $\mu(A)$, where μ is a fixed intensity measure*.*

(ii) Counts in non-overlapping regions are independent.

The intensity measure μ encodes where points are expected to be dense or sparse: a region with large $\mu(A)$ will receive many points on average. If $\mu(A) = \int_A \lambda(x)\,dx$ for some non-negative function λ, then λ is called the intensity function*.*

In this thesis, a directed multigraph is constructed from a PPP on $\mathbb{R}_+ \times \mathbb{R}_+$, where each point

(θ_i, θ_j) *represents a directed edge from node i to node j located at positions θ_i, θ_j on the positive real line.*

Definition 14 (Completely Random Measure (CRM))**.** *A* completely random measure *(CRM) [14] is a random object that assigns a non-negative random weight to every measurable subset of some space, with one key property: weights on non-overlapping regions are independent.*

More concretely, think of a CRM as a random weighted collection of points on $\mathbb{R}_+$:

$$W = \sum_{i=1}^{\infty} w_i \, \delta_{\theta_i},$$

where each atom has a random location $\theta_i \in \mathbb{R}_+$ (a node's "address") and a random weight $w_i > 0$ (the node's sociability). The collection of pairs (w_i, θ_i) is itself drawn from a Poisson point process with intensity $\nu(dw, d\theta) = \rho(dw)\,\lambda(d\theta)$, where λ is the Lebesgue measure and ρ is the Lévy measure *(Definition 15). We write $W \sim \mathrm{CRM}(\rho, \lambda)$.*

The integrability condition $\int_0^\infty (1-e^{-w})\rho(dw) < \infty$ ensures the total mass on any finite interval stays finite almost surely.

In this thesis, each node i is associated with a weight w_i drawn from a CRM, and edge probabilities depend on the product $w_i w_j$: two nodes are more likely to connect the larger their weights. Because the weights arise from a PPP, the model extends naturally to infinitely many nodes, with all but finitely many being effectively inactive at any given observation window.

Definition 15 (Lévy Measure)**.** *The* Lévy measure *ρ of a CRM is a non-negative measure on $(0, \infty)$ satisfying*

$$\int_0^\infty \left(1 - e^{-w}\right) \rho(dw) < \infty.$$

Intuitively, $\rho(dw)$ describes how frequently atoms of size $\approx w$ appear: a region where ρ is large will produce many atoms of that size.

The key quantity governing graph behavior is the total mass *$\int_0^\infty \rho(dw)$:*

- *If this integral is infinite, there are infinitely many atoms (almost surely) the CRM has* infinite activity *and the resulting graph is sparse.*
- *If finite, there are finitely many atoms and the graph is dense.*

The Lévy measure is therefore the primary lever for controlling sparsity and the shape of the degree distribution.

Definition 16 (Generalized Gamma Process (GGP))**.** *The* Generalized Gamma Process GGP(α, σ, τ) *is the specific CRM used in this thesis as a prior over node sociability weights. Its Lévy measure is:*

$$\rho_0(dw) = \frac{\alpha}{\Gamma(1-\sigma)}\, w^{-1-\sigma}\, e^{-\tau w}\, dw,$$

where the parameters satisfy either $(\sigma, \tau) \in (-\infty, 0] \times (0, \infty)$ *or* $(\sigma, \tau) \in (0, 1) \times [0, \infty)$*, with* $\alpha > 0$.

The GGP is the generative prior for the base sociability weights w_{0i} *in the dynSNetOC model. Because exact sampling from the GGP requires infinitely many draws, this thesis instead uses the exponentially tilted BFRY (etBFRY) distribution of Lee et al. [15] as a finite, convergent approximation (see Section 3.1).*

2.1.3 Deep Learning

Here, we give some definitions in the area of Deep Learning that are adjacent to the work in the Thesis.

Definition 17 (Permutation Invariance)**.** *Let* $\mathbf{X} \in \mathbb{R}^{N\times d}$ *be a matrix of* N *node feature vectors. A function* $f : \mathbb{R}^{N\times d} \to \mathbb{R}^k$ *is* permutation invariant *if, for every permutation matrix* $\mathbf{P} \in \{0,1\}^{N\times N}$,

$$f(\mathbf{P}\mathbf{X}) = f(\mathbf{X}).$$

Intuitively, the output of f *does not depend on the order in which nodes are presented. Graph-level predictions (e.g. graph classification, total edge count) should be permutation invariant, since the abstract graph structure is independent of node labelling.*

Definition 18 (Permutation Equivariance)**.** *A function* $f : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times k}$ *is* permutation equivariant *if, for every permutation matrix* $\mathbf{P}$,

$$f(\mathbf{P}\mathbf{X}) = \mathbf{P}\, f(\mathbf{X}).$$

That is, permuting the input nodes permutes the output in the same way. Node-level predictions (e.g. community affiliations, node embeddings) should be permutation equivariant: if node i *is relabelled, its predicted representation should simply move to the new label, not change in value.*

Definition 19 (Message Passing Neural Network (MPNN))**.** *A Message Passing Neural Network [16] is a class of graph neural network in which each node* v *maintains a hidden representation* $\mathbf{h}_v^{(l)} \in \mathbb{R}^d$ *that is updated over* L *rounds of* message passing. *At each round* l*, every* *node aggregates information from its neighbourhood $\mathcal{N}(v)$ and updates its own state:*

$$\mathbf{m}_v^{(l)} = \text{AGGREGATE}^{(l)}\left(\left\{\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\right\}\right),$$
$$\mathbf{h}_v^{(l)} = \text{UPDATE}^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)}\right),$$

where AGGREGATE *is a permutation-invariant function (e.g. sum, mean, or max) and* UPDATE *is a learned transformation (e.g. a linear layer followed by a non-linearity). After L rounds, $\mathbf{h}_v^{(L)}$ encodes structural information from the L-hop neighbourhood of v. The map from node features $\mathbf{X}$ to updated representations $\mathbf{H}^{(L)}$ is permutation equivariant by construction, since the aggregation step is order-agnostic.*

2.2 Related Work

A wide array of model-based approaches have been developed for community detection. In this section, we will explore a subset of these methods. To explore these topics on a deeper level, we recommend [17] [18].

2.2.1 Stochastic Block Models

Stochastic Block Models (SBMs), first introduced in [4], have been extensively studied in the literature [19] [20] [21]. SBMs operate under the assumption that nodes belong to latent communities or blocks, with edges generated probabilistically according to group memberships. The probability of a node belonging to a community i connecting with another node belonging to a community j is given by a block-matrix **B**, where $\mathbf{B}_{ij} \in [0, 1]$.

Yang et al. [22] extended the standard SBM to model dynamic graphs by allowing group memberships to evolve through time based on a Markov chain, in this model the block-matrix **B** remains constant through time. Xu and Hero [23] extended this work by evolving both memberships and **B** through time. For the evolution of **B** the authors use a state-space model on the logistic transformation of the probabilities of the matrix while still allowing group memberships to evolve through time. The approach described in [23] suffers from label switching, where labels assigned to these groups can be arbitrarily swapped between two successive timesteps without changing the model's fit to the data, making the interpretation of communities difficult. Matias and Mielle [7] proposed a solution to this issue by adding an additional constraint to their model(dynSBM), that the connectivity behavior within each group must remain stable over time. Through experimental results, it is shown that the model no longer suffers from label switching.

Another extension to the classical SBM framework is the mixed membership stochastic block model (MMSB) proposed by Airoldi et al. [5], which enables overlapping community detection by relaxing the constraint that a node is affiliated with only one community and assigning each node a membership vector that represents its degree of participation across multiple communities. Building on this work, Xing et al. [6] extended the MMSB framework to accommodate dynamic networks by incorporating a state space model that allows membership vectors to evolve over time, thereby enabling overlapping community detection in temporal graphs. In general, SBM-based approaches suffer from an inability to capture sparsity and cannot explicitly model power-law degree distributions.

2.2.2 Latent Multiplicative Models

Latent multiplicative models are a class of latent space models, which assume the probability of an edge between two nodes depends on their positions in an unobserved "social space". The initial work by Hoff et al [24], proposed two approaches: distance models and projection models. While distance models assume that the probability of a connection is related to the distance of two nodes in latent space, projection models are the framework of this multiplicative approach. In a projection model, the log-odds of observing a node between two nodes is given by:

$$logit(A_{ij}) = \alpha + \frac{z_i^T z_j}{|z_j|}$$

Where $z_i = a_i v_i$. Here, a acts as a sociability parameter and v_i is a K-dimensional vector acting as the node affiliation to each community.

This concept was further developed by Hoff [25] into the generalized bilinear mixed-effects (GBME) model. This models represents the latent structure using sender-specific(u_i) and receiver-specific(v_j) latent factors with the log-odds being defined as:

$$logit(A_{ij}) = u_i^T D v_j + \epsilon_{ij}$$

The approach follows the intuition of Singular Value Decomposition (SVD), where the interaction matrix is approximated by a low-rank representation capturing the main latent dimensions contributing to the adjacency matrix.

Several extensions have been made to the multiplicative framework for dynamic networks. Ward et al [26], decided to extend the GBME by using the fitted latent positions from the previous timestep $t-1$ as predictors for the next timestep t. In contrast, Durante and Dunson [27] proposed a dynamic bilinear effects model in which latent factors z_{ijt} themselves evolve over time according to Gaussian processes (GPs). This GP-based approach has the advantage of relaxing the strict Markovian assumption common in other dynamic models and can naturally handle time points that are not equally spaced.

A similar approach to GBME is the one described in Psorakis et al. [28] where the problem is approached as a Bayesian Nonnegative Matrix Factorization (NMF) task instead of a Singular Value Decomposition(SVD). This framework poses the generative assumption that the observed adjacency matrix A is approximated by the product of two non-negative factor matrices, $A \approx \mathbf{WH}$. In this model, the matrix $\mathbf{W}$ represents the node-community assignment weights, effectively providing a soft-membership vector for each node, while $\mathbf{H}$ contributes to reconstructing the expected interaction strengths.

2.2.3 Deep Learning Methods

Deep learning-based methods have recently gained attention for their ability to capture complex, nonlinear network structures that are often missed by traditional linear models. These approaches typically leverage deep neural networks to learn low-dimensional node representations, or embeddings, that are optimized for community detection tasks.

A prominent approach in this area utilizes autoencoders [29] to learn nonlinear representations for network reconstructions. Yang et al. [30] argue that stochastic models can be viewed as finding low-dimensional linear embeddings to reconstruct a network's adjacency. To overcome the limitations of linearity, they propose a Deep Nonlinear Reconstruction (DNR) model that uses a stacked autoencoder to learn a powerful nonlinear embedding of the modularity matrix, from which communities are then identified. Building on this concept for temporal graphs, Wang et al. [31] introduce a semi-supervised evolutionary autoencoder (sE-AE). Their model is situated within an evolutionary clustering framework which seeks to balance the quality of the community structure at a given timestep with the temporal smoothness between consecutive steps. Using an autoencoder, the model can effectively capture the nonlinear properties of dynamic networks.

More recently, Graph Neural Networks (GNNs), which are specifically designed to operate on graph-structured data, have been applied to community detection. Shchur and Günnemann [10] proposed a model (NOCD) that utilizes the node embeddings generated by a GNN along with a Bernoulli-Poisson probabilistic model (something we will explore further in Section 2.2.4), which naturally allows for overlapping memberships, for the generation process of a graph, the node embeddings can then be used as the mixed affiliation vectors of each node. A deeper explanation of the NOCD model can be found in Section 4.1.2.

It is important to note that unlike more classical methods, Deep Learning methods, provide expressiveness at the cost of interpretability, something that might be important in specific applications. This is due to the black-box nature of these models.

2.2.4 Caron-Fox Family of Models

Thus far, the methods we discussed do not explicitly model sparsity and cannot capture the power-law degree distribution that many real-world graph datasets exhibit. In general, exchangeability is an important attribute that these models must exhibit. It has been shown [32], that exchangeability in Bayesian models like [5][7][23][6] can only produce dense graphs and cannot model sparsity. To tackle this issue, Caron and Fox [8], proposed an alternative way to view graphs, as an infinite point process on a plane $\mathbb{R}_+^2$.

$$Z = \sum_{i,j} z_{ij} \delta(\theta_i, \theta_j) \tag{2.1}$$

where $z_{ij} = 1$ if there is a link between nodes θ_i and θ_j in $\mathbb{R}^+$, and is 0 otherwise. To avoid self-edges, where a node connects with itself, the authors added the $\delta(\theta_i, \theta_j)$ term, where δ is a Kronecker delta specifying that if $\theta_i = \theta_j$ then $\delta(\theta_i, \theta_j)$ is 1 otherwise it is 0.
This formulation allows models to capture an infinite number of unobserved nodes enabling a Bayesian nonparametric approach to graph modeling. The authors then go on to describe a generative model to construct the graph:

$$Pr(z_{ij} = 1 | w_i, w_j) = 1 - e^{-2w_i w_j} \tag{2.2}$$

The weights w_i can be interpreted as measuring the sociability of a node i in the graph. These weights are drawn from a Poisson point process (Definition 13) with a multivariate Lévy measure (Definition 15), forming a vector of completely random measures (CRMs) (Definition 14). Theoretical analysis and experimental results demonstrate that this model can generate sparse graphs. Crucially, by choosing appropriate Lévy measures, the model can produce power-law degree distributions, addressing a key limitation of most other methods for community detection.

Herlau et al. [33] built on top of this framework by re-introducing latent block-structure(from SBMs) to the Caron-Fox model , which the original model did not capture. This resulted in a model capable of inferring block-structures, essentially allowing it to model non-overlapping communities, while also capturing edge heterogeneity, such as power-law degree distributions.

Todeschini et al. [9] also extended the model by allowing it to model overlapping communities on graphs. This was done by reformulating the generative model like so:

$$Pr(z_{ij} = 1 | (w_{l1}, ..., w_{lp}) = 1 - e^{-2\sum_{k=1}^{p} w_{ik} w_{jk}}$$

where the weights w_{ik} can now be interpreted as measuring the level of affiliation of a node i a community k. Another extension to the standard Caron-Fox model was to generalize the model to work on dynamic graphs by Naik et al. [34], essentially allowing for the evolution of weights through time. The evolution is modeled as a time-varying generalized gamma process allowing for interpretation of the temporal evolution of the weights.

It should be emphasized that none of the aforementioned methods perform overlapping community detection on dynamic graphs. In this work, we build upon the framework of Todeschini et al. [9] by extending the model to handle temporal graphs, enabling the capture of both community structure and its evolution over time.

In Table 2.1, we provide a summary of the capabilities of each model described in this section.

	Features			
Models	Overlapping	Dynamic	Sparsity	Degree Hetero-geneity
SBM[4]	✗	✗	✗	✗
dSBM[22]	✗	✓	✗	✗
dySBM[23]	✗	✓	✗	✗
dynSBM[7]	✗	✓	✗	✗
MMSB[5]	✓	✗	✗	✗
dMMSB[6]	✓	✓	✗	✗
LFM[24]	✓	✗	✗	✗
GBME[25]	✓	✗	✗	✗
TGBME[26]	✓	✓	✗	✗
GPGBME[27]	✓	✓	✗	✗
BNMF[28]	✓	✗	✗	✗
AECD[30]	✓	✗	✗	✗
sE-AE[31]	✓	✓	✗	✗
NOCD[10]	✓	✗	✗	✗
CF[8]	✗	✗	✓	✓
SnetDC[33]	✗	✗	✓	✓
SnetOC[9]	✓	✗	✓	✓
dSnet[34]	✗	✓	✓	✓
dynSNetOC	✓	✓	✓	✓
dNOCD	✓	✓	✓	✗

Table 2.1: Comparison of different features between community detection models. **dynSNetOC** and **dNOCD** are the proposed models of the thesis. We compare their capabilities with all other models mentioned in Related Work.

2.3 Tools

This section briefly describes the main software libraries and computational frameworks used to implement the models and experiments presented in this thesis.

2.3.1 JAX

JAX [35] is a high-performance numerical computing libraryfor Python, developed by Google Research. It combines a familiar NumPy-like API (jax.numpy) with a powerful set of function transformations, including Just-In-Time (JIT) compilation (jax.jit), automatic differentiation (jax.grad), and vectorization/parallelization (jax.vmap, jax.pmap). JAX can run code transparently on CPUs, GPUs, and TPUs. In this work, JAX serves as the fundamental backend for all probabilistic models. Its ability to JIT-compile entire computation graphs provides significant performance speedups, while its automatic differentiation is essential for gradient-based inference algorithms, such as those used in HMC and NUTS.

2.3.2 Numpyro

Numpyro [36,37] is a lightweight probabilistic programming library (PPL) built on top of JAX. It provides a simple and flexible interface for defining probabilistic models as Python programs, and it leverages JAX for compilation and execution. Numpyro offers a wide range of inference algorithms, including Hamiltonian Monte Carlo (HMC), the No-U-Turn Sampler (NUTS), and Stochastic Variational Inference (SVI). We use Numpyro as the primary framework for implementing, fitting, and performing inference on the Bayesian models described in this thesis, particularly the extension to the Caron-Fox family (Section 2.2.4). The combination of Numpyro's expressive modeling language and JAX's high-performance backend allows for efficient and scalable inference on complex generative models.

2.3.3 PyTorch

PyTorch [38] is a widely-adopted open-source machine learning library primarily used for deep learning applications. It is known for its flexibility, imperative programming style (eager execution), and strong GPU acceleration for tensor computations. PyTorch also has an extensive ecosystem, including libraries like PyTorch Geometric (PyG) [39,40], which is specifically designed for implementing Graph Neural Networks (GNNs). In this thesis we leverage the PyTorch ecosystem to define and train graph neural networks for community detection.

2.3.4 Computational Resources

The experiments in this thesis were carried out on two hardware environments, chosen according to the demands of each model.

The Bayesian model of Chapter 3 was developed on an Apple M1 MacBook. For the larger and more computationally demanding runs, we utilized the High-Performance Computing (HPC) facility of the University of Cyprus[1]. These experiments were executed on cluster nodes equipped with NVIDIA V100 GPUs. The university's HPC infrastructure provides high computational power, featuring 3,000 processor cores, 16 NVIDIA V100 graphics cards, 15 TB of RAM. In total, approximately 168 GPU hours were consumed on the HPC cluster for the purpose of this research. We gratefully acknowledge the University of Cyprus for providing these invaluable computational resources and services free of charge.

The deep learning model of Chapter 4 was developed and run entirely on an M1 MacBook, as it did not require additional computational resources.

[1]https://hpc.ucy.ac.cy

3 Bayesian Network Modeling

TABLE OF CONTENTS

In this chapter we describe a Bayesian model for overlapping community detection of dynamic sparse graphs. Building such a model is non-trivial because four desirable properties must hold simultaneously: *sparsity & degree heterogeneity*, since many observed networks are sparse and exhibit power-law degree distributions; *overlapping community structure*, since nodes typically belong to multiple groups to varying degrees; and *temporal evolution*, since both connectivity and community affiliations change over time. Existing dynamic stochastic block models rely on exchangeable adjacency-matrix representations, which forces the resulting graphs to be either empty or dense [32], and therefore cannot accommodate the first property. Our model captures sparsity and power-law degree distributions by formulating the problem as a point process, following Caron & Fox [8]. Overlapping community affiliation is captured by assigning to each node a sociability vector β_{ik}, following Todeschini et al. [9]. We extend this prior work by allowing community affiliations to evolve over time through a latent Markov process, yielding a genuinely dynamic generative model rather than a sequence of independent static snapshots.

The remainder of the chapter is organized as follows. Section 3.1 defines the generative model and discusses the role of each parameter. Section 3.2 describes the inference procedure and Section 3.3 the optimizations that make it computationally tractable. Section 3.4 describes the experimental methodology and baseline comparison with other overlapping community models.

3.1 Model Definition

We model a sequence of T undirected multigraphs on a shared node set of size L. The generative process is as follows:

- *Base sociability weights (degree correction):*
  For each node $i = 1, \ldots, L$:
  - $w_{0i} \sim \text{etBFRY}(\alpha, \tau, \sigma)$, sample the base sociability of node i.
- *State-space model for community affiliation scores:*
  For each node $i = 1, \ldots, L$ and community $k = 1, \ldots, p$,
  - $\beta_{ik}^{(1)} \sim \text{Gamma}(a_k, b_k)$, sample the initial affiliation score of node i to community k at time 1.

  For $t = 2, \ldots, T$:
  - $\gamma_{ik}^{(t-1)} \mid \beta_{ik}^{(t-1)} \sim \text{Gamma}(\psi, \beta_{ik}^{(t-1)})$, sample the auxiliary transition variable.
  - $\beta_{ik}^{(t)} \mid \gamma_{ik}^{(t-1)} \sim \text{Gamma}(a_k + \psi,\ b_k + \gamma_{ik}^{(t-1)})$, sample the affiliation score at time t.
  - $w_{ik}^{(t)} = w_{0i}\, \beta_{ik}^{(t)}$, compute the sociability weight of node i in community k at time t.
- *Compound completely random measure for networks:*
  For each pair of nodes $(i, j) \in [1, L] \times [1, L]$, at each time point $t = 1, \ldots, T$:
  - For a binary graph:
    - $z_{ij}^{(t)} \mid (w_{ik}^{(t)}, w_{jk}^{(t)}) \sim \text{Bernoulli}(1 - \exp(-2 \sum_{k=1}^{p} w_{ik}^{(t)} w_{jk}^{(t)}))$, sample binary edges between nodes i and j.
  - For a multigraph:
    - $n_{ij}^{(t)} \mid (w_{ik}^{(t)}, w_{jk}^{(t)}) \sim \text{Poisson}(\sum_{k=1}^{p} w_{ik}^{(t)} w_{jk}^{(t)})$, sample multi-edges between nodes i and j.

A graphical representation of the model can be seen in Figure 3.1.

The priors $\alpha, \sigma, \tau, \psi, a, b$ are sampled from vague distributions on their respective domains: $\alpha, \tau, \psi > 0$, $\sigma \in (0, 1)$, and $a_k, b_k > 0$ for $k = 1, \ldots, p$.

The base sociability w_{0i} is drawn from an exponentially tilted etBFRY distribution, a finite-dimensional approximation to the generalized gamma process introduced in Lee et al. [15]. This approximation preserves the key structural properties of the underlying process, namely sparsity and power-law degree behavior, while keeping the parameter space finite and inference computationally tractable.

Parameter Interpretation. The hyperparameters α, σ, τ control the global properties of the graph: α governs its overall size, σ controls sparsity and the power-law exponent of the degree

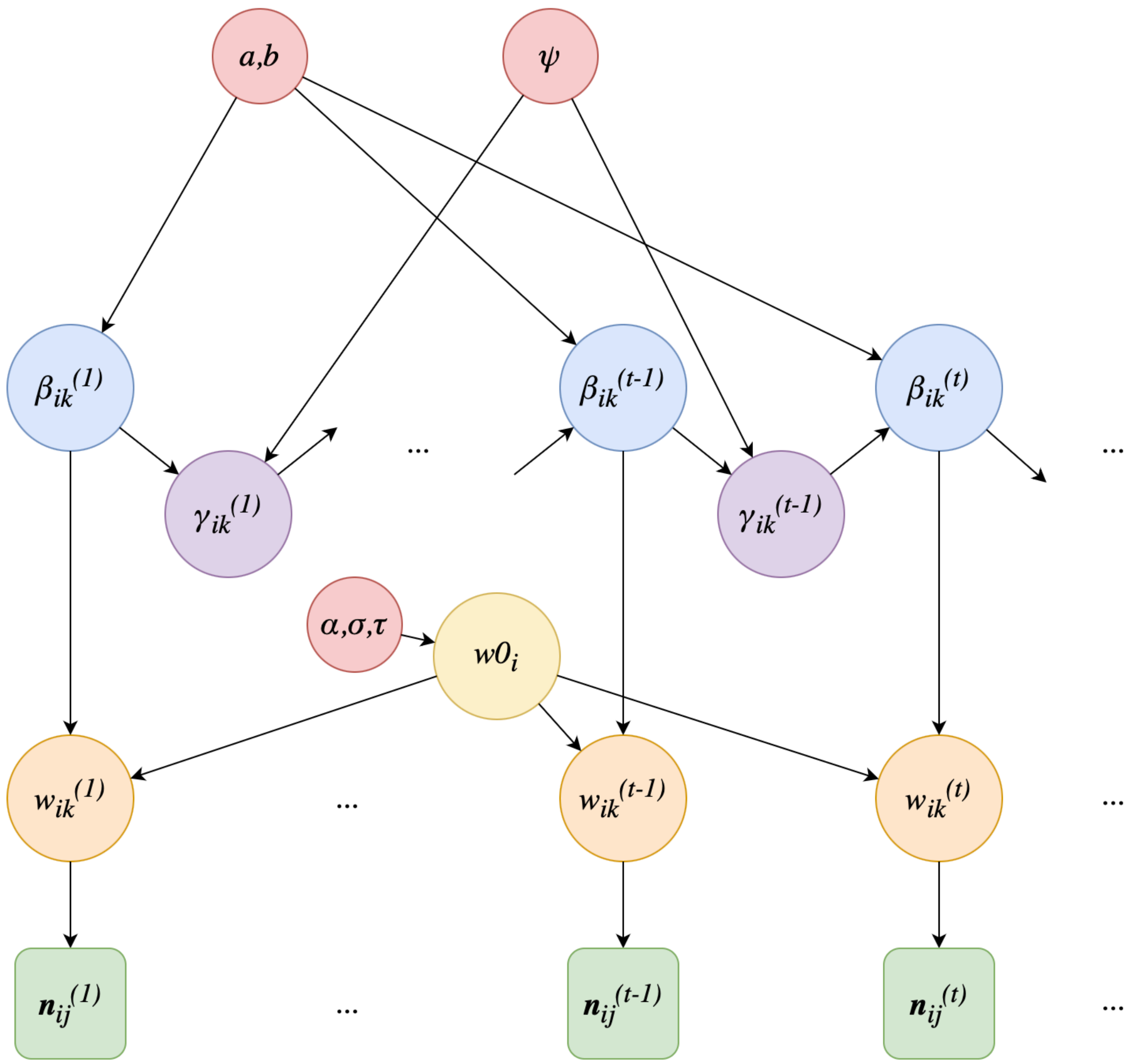

Figure 3.1: A visualization of the latent structure of dynSNetOC.

distribution, and τ acts as an exponential tilting parameter. The vectors $a = (a_1, \ldots, a_p)$ and $b = (b_1, \ldots, b_p)$ control the distribution of affiliation scores within each community, while w_{0i} acts as a node-specific degree-correction parameter that absorbs the sparsity and degree heterogeneity. The community-specific scores $\beta_{ik}^{(t)}$ represent the strength of node i's affiliation to community k at time t, and since these vectors essentially capture overlapping memberships for each node. The temporal coupling parameter ψ governs the strength of correlation between successive timesteps: large values of ψ produce smooth, slowly evolving community structure, while small values allow rapid turnover.

A useful structural property of the latent Markov process is that it is stationary by construction: the marginal distribution of $\beta_{ik}^{(t)}$ remains Gamma(a_k, b_k) for every t, so the model has no preferred direction in time and the global properties of the graph are preserved across timesteps. Finally, the rate $w_{ik}^{(t)} w_{jk}^{(t)}$ governing the Poisson draw of edges within community k produces a non-negative Poisson factorization across communities [9], which is precisely what enables a single observed edge between i and j to be explained by contributions from multiple communities simultaneously.

3.2 Inference

Given an observed sequence of graphs $\{n^{(t)}\}_{t=1}^{T}$ on L nodes, our goal is to recover the latent sociability weights w_{0i}, the time-varying community affiliation scores $\beta_{ik}^{(t)}$, the auxiliary transition variables $\gamma_{ik}^{(t-1)}$, and the hyperparameters $\xi = (\alpha, \sigma, \tau, \psi, a, b)$. The joint posterior is analytically intractable due to the coupling induced by the latent Markov process and the Poisson likelihood over all node pairs. We therefore resort to approximate Bayesian inference using gradient-based Markov chain Monte Carlo. Specifically, we use the No-U-Turn Sampler (NUTS) [13], an adaptive variant of Hamiltonian Monte Carlo, as implemented in the probabilistic programming language NumPyro [36,37]. NUTS is well suited to the high-dimensional continuous parameter space induced by the dynamic latent variables, since it exploits gradient information to traverse the posterior efficiently and adapts the trajectory length automatically. The model is specified declaratively in NumPyro, and gradients are obtained through JAX's automatic differentiation, allowing the entire posterior geometry to be explored without hand-derived updates.

3.3 Optimizations

A naive implementation of the model is computationally expensive to infer. Sparsity and degree heterogeneity are properties best examined at scale, and most real-world graph datasets are large, so a direct implementation quickly becomes burdensome to run. Identifying optimizations that exploit the structure of the model is therefore essential to making it practical for analysis.

Sparse Poisson likelihood. A naive evaluation of the Poisson log-likelihood at each timestep requires computing the full $L \times L$ rate matrix $\Lambda_{ij}^{(t)} = \sum_{k=1}^{p} w_{ik}^{(t)} w_{jk}^{(t)}$, costing $O(L^2 p)$ per timestep. This is wasteful when the observed graph is sparse, since the vast majority of entries are zero and contribute only the $\Lambda_{ij}^{(t)}$ term to the Poisson log-density. We exploit this by splitting the log-likelihood into two parts. The aggregate term $\sum_{i,j} \Lambda_{ij}^{(t)}$ can be computed in closed form using the identity

$$\sum_{i,j} \sum_{k=1}^{p} w_{ik}^{(t)} w_{jk}^{(t)} = \sum_{k=1}^{p} \Big(\sum_{i=1}^{L} w_{ik}^{(t)} \Big)^2, \tag{3.1}$$

which collapses the double sum over node pairs into p squared column sums, costing only

$O(Lp)$. The remaining $n_{ij}^{(t)} \log \Lambda_{ij}^{(t)}$ terms are nonzero only at observed edges, and are therefore evaluated solely at the $\mathrm{nnz}^{(t)}$ non-zero entries of the adjacency matrix at cost $O(\mathrm{nnz}^{(t)} \cdot p)$. The total per-timestep cost drops from $O(L^2 p)$ to $O((L + \mathrm{nnz}^{(t)}) \cdot p)$, which for sparse networks with $\mathrm{nnz} = O(L)$ is a substantial reduction. The per-timestep evaluations are further vectorized across t using JAX's `vmap`, so the entire likelihood is computed in a single fused operation.

Non-centered reparameterization of the latent Markov process. The natural specification of the latent Markov process draws the transition variable as $\gamma_{ik}^{(t-1)} \mid \beta_{ik}^{(t-1)} \sim \mathrm{Gamma}(\psi, \beta_{ik}^{(t-1)})$ and the affiliation score as $\beta_{ik}^{(t)} \mid \gamma_{ik}^{(t-1)} \sim \mathrm{Gamma}(a_k + \psi,\, b_k + \gamma_{ik}^{(t-1)})$, both with rate parameters that depend on other latent variables. These data-dependent rates create strong posterior coupling between successive timesteps, producing a poorly conditioned posterior geometry that NUTS struggles to traverse efficiently. We exploit the scaling property of the Gamma distribution to apply a non-centered reparameterization to both $\gamma_{ik}^{(t-1)}$ and $\beta_{ik}^{(t)}$: each variable is sampled from a Gamma distribution with a fixed unit rate, and then rescaled deterministically by the appropriate rate parameter. The resulting distributions are identical to the original specification, but the sampled variables now have posterior geometries that are decoupled from the conditioning latent variables. This yields markedly improved sampling efficiency in NUTS, with higher effective sample sizes per iteration and fewer divergent transitions.

3.4 Experiments and Model Comparisons

In this section we evaluate the proposed dynSNetOC model on both synthetic and real-world data. The synthetic experiment in Section 3.4.1 validates that the approximate inference procedure recovers the ground-truth parameters of a graph generated from the model itself. The real-world experiments in Sections 3.5.2 and 3.5.3 apply the model to the Reuters Terror dataset and a European Flights dataset respectively to test the model's ability to create thematically consistent community representations and compare them against the baseline models described in Section 3.4.3.

3.4.1 Synthetic Evaluation

The purpose of the synthetic experiment is to verify that the approximate inference procedure of Section 3.2, together with the optimizations of Section 3.3, are capable of recovering the latent structure of a graph drawn from the dynSNetOC generative process itself. Because we know the true sociabilities, scores, and hyperparameters that produced the data, this controlled setting lets us measure inference accuracy directly, something that is not possible to do with the real-world datasets.

We generate a single dynamic undirected multigraph from the generative process of Section 3.1

with $T = 6$ timesteps and $p = 2$ communities, using fixed hyperparameter values $\alpha = 50$, $\sigma = 0.2$, $\tau = 1$, $\psi = 5$, $a = (1, 1)$, $b = (1, 2)$. A sample graph generated from this process can be seen in Figure 3.2.

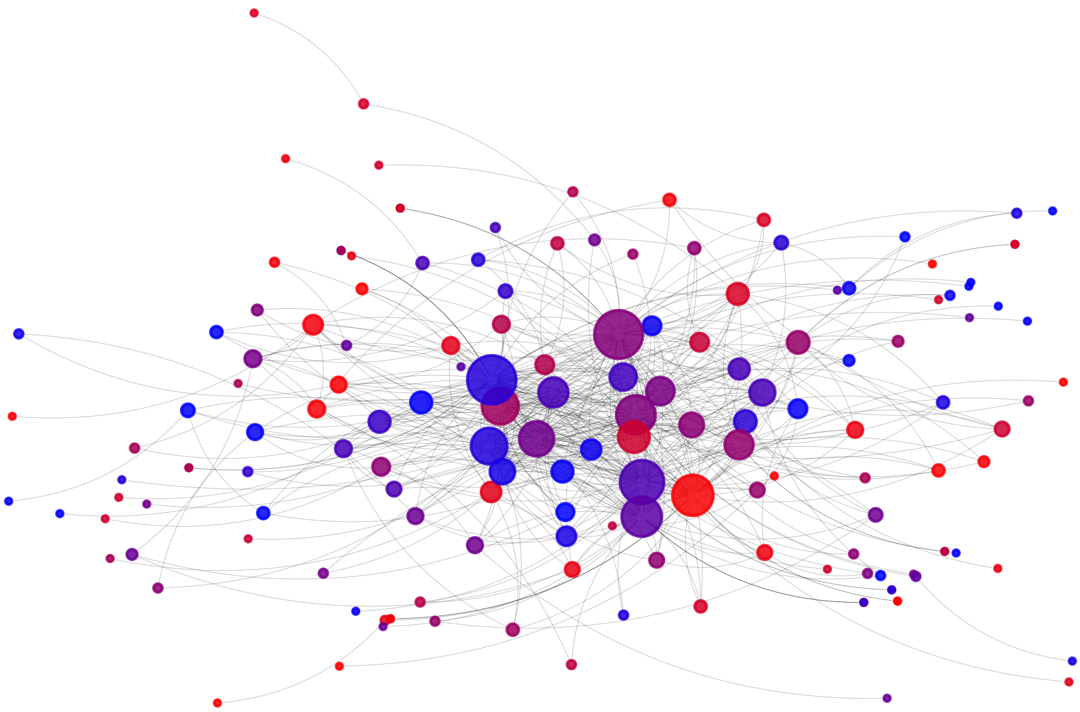

Figure 3.2: Graph visualization of the synthetic dataset at a single timestep, generated from the dynSNetOC model with p = 2 communities. Node color reflects community affiliation, ranging from red (community 1) to blue (community 2), with purple indicating mixed membership. Node size is proportional to the degree of each node.

3.4.2 Datasets

To evaluate the model in real-world settings, we apply it to two distinct datasets that exhibit contrasting structural behaviors: a dynamic word co-occurrence network derived from the Reuters Terror dataset, and a European flights network dataset. The contrast between the two allows us to stress-test the model across qualitatively different regimes.

3.4.2.1 Reuters Terror Dataset

The Reuters Terror dataset[1] consists of articles published by Reuters News Agency between September 11 and November 15, 2001, covering the immediate aftermath of the 9/11 attacks.

[1] https://sparse.tamu.edu/Pajek

Each article is represented as a word co-occurrence network, where **nodes** correspond to words and **edges** connect words that appear in the same sentence. The full corpus is aggregated into six weekly snapshots, each capturing the co-occurrence structure of that week's articles as a single graph. A visual representation of a sample week can be seen in Figure 3.3.

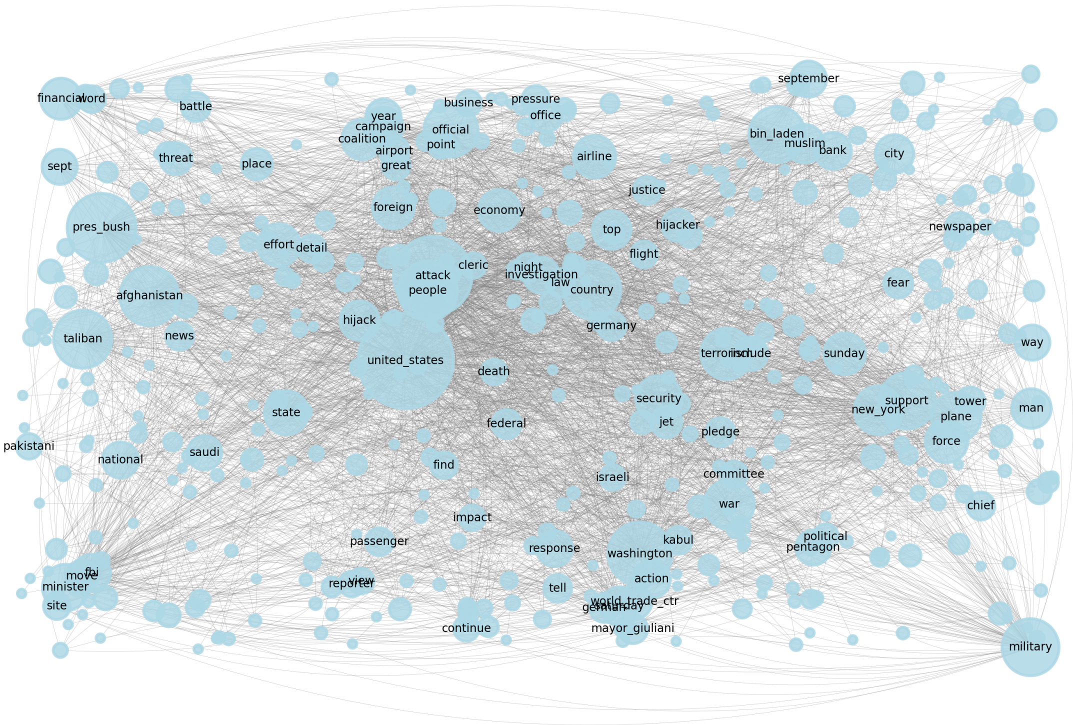

Figure 3.3: Graph visualization of the Reuters Terror word co-occurrence network at a single timestep. Each node represents a word, with node size proportional to its degree (i.e., frequency of co-occurrence with other words).

The dataset exhibits several favorable properties for our model. The network consists of $L = 1200$ nodes in total, though not all words are present at every week: on average only around 50% of nodes are active at each timestep, with active node counts of $(468, 525, 501, 474, 471, 525)$ across the six weeks. The graphs are sparse, with empirical sparsity (measured as the number of edges divided by the square of the number of active nodes) of $(0.02, 0.01, 0.01, 0.02, 0.02, 0.01)$ at each timestep. Furthermore, the degree distribution at each timestep is heavy-tailed and well described by a power law, producing linear log-log degree distribution plots as shown in Figure 3.4. Connectivity also changes at every timestep, reflecting the evolving focus of news coverage over the weeks following the attack. Together, these properties make the dataset a natural fit for dynSNetOC, which is specifically designed to handle sparse, power-law networks with dynamic overlapping community structure.

A well fitted model should be able to recover both the degree heterogeneity of the network and create semantically consistent overlapping communities throughout the weeks.

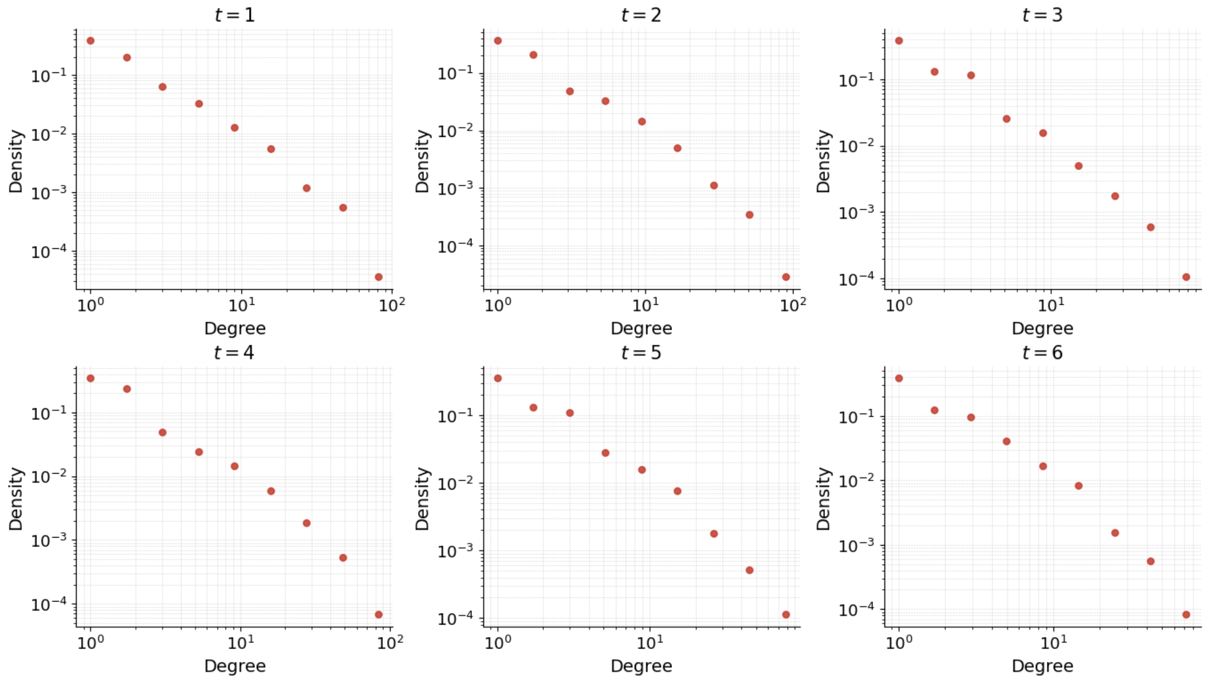

Figure 3.4: Degree distribution in log-log scale for all timesteps of the Reuters Terror co-occurrence network.

3.4.2.2 European Flights Dataset

The European Flights dataset is constructed from the EUROCONTROL Aviation Data Repository for Research[2]. The repository contains all historic commercial flights (IFR, scheduled and non-scheduled, excluding military, state and general aviation) recorded by the EUROCONTROL Network Manager for four fixed sample months per year (March, June, September and December). A visual representation of a sample month can be seen in Figure 3.5 We use the four months available for 2023, giving $T = 4$ snapshots, one per month. **Nodes** are airports, identified by their ICAO code, and an **edge** between two airports at time t represents a flight with that pair as departure (ADEP) and destination (ADES) airports in month t, creating an undirected multigraph. Not all airports are active in every month, and only around 35% of the nodes are active per timestep. Empirical sparsity at each timestep is $(0.034, 0.037, 0.039, 0.035)$. The data consists of 4 graphs of $L = 1500$. The empirical degree distribution in each time is power-law, giving a linear log-log degree distribution that can be seen in Figure 3.6.

[2] EUROCONTROL Aviation Data Repository for Research, available at https://www.eurocontrol.int/dashboard/aviation-data-research

Figure 3.5: Visualization of the European Flights network for a sample month overlayed on a map of the earth (Top) and a zoomed in version showing Europe only (Bottom). Each node represents an airport, with node size proportional to its degree (i.e. the amount of outbound/inbound flights of the airport)

3.4.3 Baseline Models

To assess the strengths of dynSNetOC, we compare it against two established dynamic community detection models: the dynamic Mixed Membership Stochastic Blockmodel (dMMSB) of Xing et al. [6], which captures overlapping memberships but assumes a dense exchangeable graph, and the dynamic Stochastic Blockmodel (dynSBM) of Matias & Miele [7], which captures temporal evolution but restricts nodes to a single community at each timestep. Neither model accommodates sparsity, power-law degree distributions, and overlapping communities

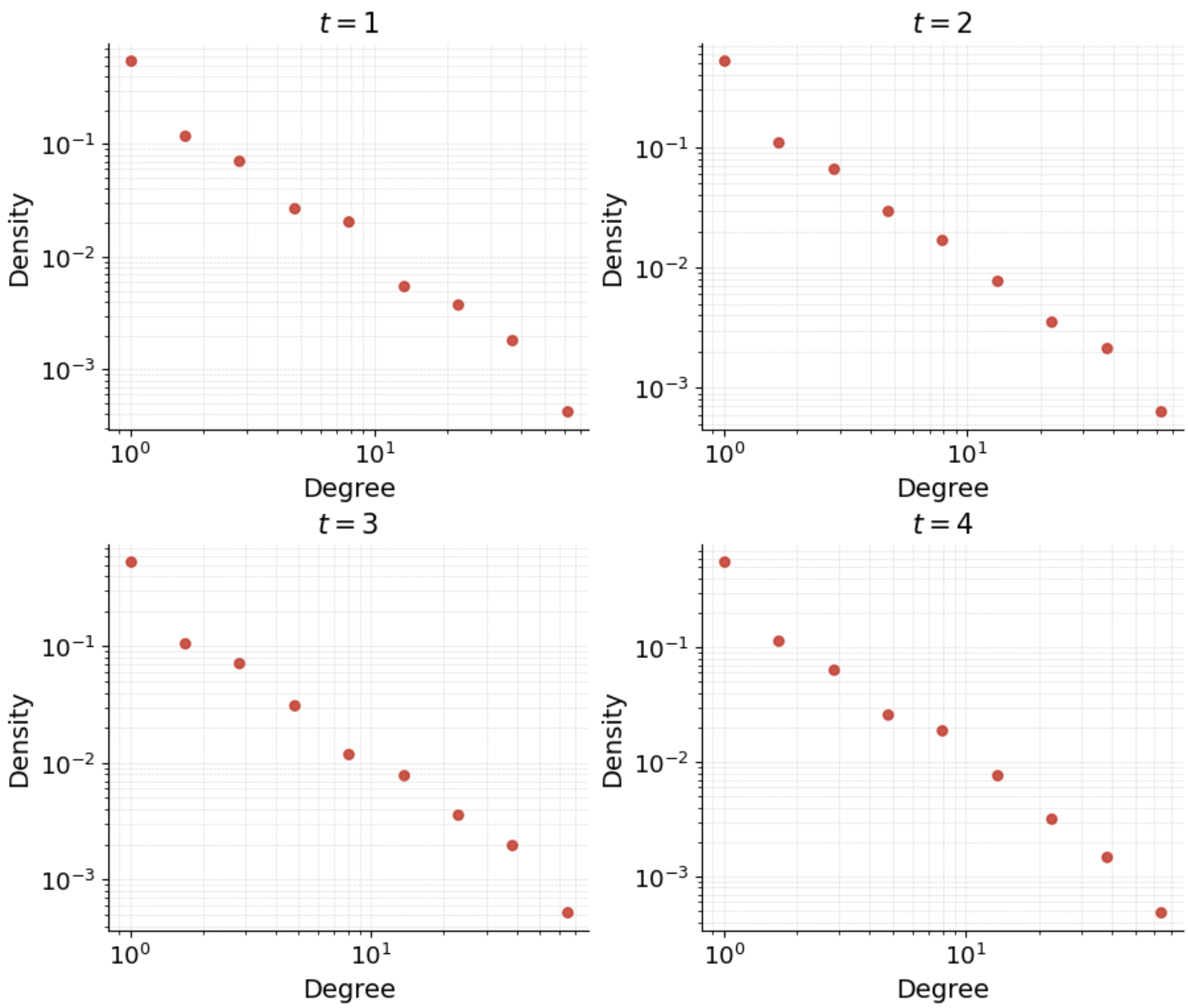

Figure 3.6: Degree distribution in log-log scale for all timesteps of the European Flights network.

simultaneously, and the comparison is meant to highlight the gap our model fills.

3.4.3.1 dMMSB

The dynamic Mixed Membership Stochastic Blockmodel of Xing et al. [6] describes both static and dynamic versions of the model, together with a variational inference algorithm. A mixed-membership vector is drawn for each node from a Logistic Normal distribution and evolved over time via a state-space model; links between nodes are then generated using a role-compatibility matrix. A visual representation of the model can be found in Figure 3.7.

Let $\{G_t = (V, E_t)\}_{t=1}^T$ denote a sequence of graphs on a set of N nodes, with symmetric adjacency matrices $z^{(t)} = [z_{ij}^{(t)}]_{i,j=1}^N$ (no self-loops) and let $\mu^{(t)} = (\mu_1^{(t)}, \dots, \mu_p^{(t)})$ be the mean of the mixed membership prior at time t, and let $\eta_{rq}^{(t)}$ denote the role-compatibility parameter between roles r and q at time t. We define the generative model as:

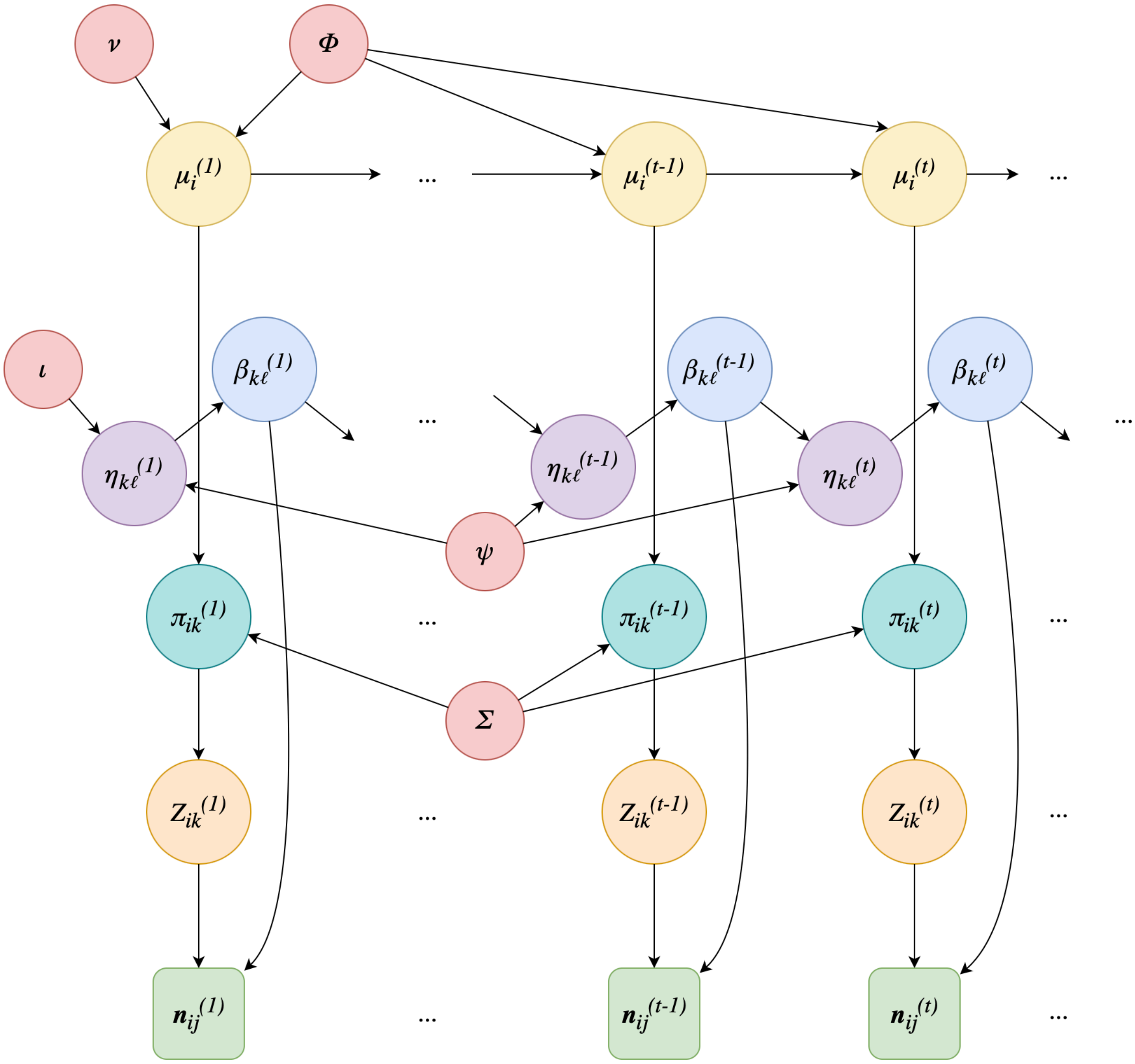

Figure 3.7: Graphical visualization of the dMMSB model.

- *State-space model for the mixed membership prior mean:*
  - $\mu^{(1)} \sim \text{Normal}(\nu, \Phi)$, sample the mean of the mixed membership prior at time 1.

  For $t = 2, \ldots, T$:
  - $\mu^{(t)} \mid \mu^{(t-1)} \sim \text{Normal}(\mathbf{A}\mu^{(t-1)}, \Phi)$, sample the mean of the mixed membership prior at time t.
- *State-space model for the role-compatibility matrix:*

  For each pair of roles $r, q \in \{1, \ldots, p\}$:
  - $\eta_{rq}^{(1)} \sim \text{Normal}(\iota, \psi)$, sample the compatibility coefficient between roles r and q at time 1.

  For $t = 2, \ldots, T$:

- $\eta_{rq}^{(t)} \mid \eta_{rq}^{(t-1)} \sim \text{Normal}(b\eta_{rq}^{(t-1)}, \psi)$, sample the compatibility coefficient at time t.
- $\beta_{rq}^{(t)} = \exp(\eta_{rq}^{(t)})/(\exp(\eta_{rq}^{(t)}) + 1)$, compute the compatibility probability via the logistic transform.

- *Node memberships and edge generation:*
  For each node $i = 1, \dots, N$ at each time $t = 1, \dots, T$:
  - $\pi_i^{(t)} \sim \text{LogisticNormal}(\mu^{(t)}, \Sigma^{(t)})$, sample the mixed membership vector of node i.
  - $Z_i^{(t)} \mid \pi_i^{(t)} \sim \text{Multinomial}(\pi_i^{(t)}, 1)$, sample the latent group label of node i.

  For each pair of nodes $(i, j) \in [1, N] \times [1, N]$, at each time $t = 1, \dots, T$:
  - $z_{ij}^{(t)} \mid (Z_i^{(t)} = q,\ Z_j^{(t)} = \ell) \sim \text{Bernoulli}(\beta_{q\ell}^{(t)})$, sample the edge between nodes i and j.

The variational EM algorithm described in the paper assumes that the role-compatibility matrix is constant through time and that the transition matrix **A** is equal to the identity matrix **I**, which reduces the state-space model to a random walk.

Since no public implementation of the algorithm was available, we implemented the paper from scratch and made the code publicly available[3].

3.4.3.2 dynSBM

The dynamic Stochastic Blockmodel of Matias & Miele [7] extends the classical SBM by letting each node's single community membership evolve over time according to a first-order Markov chain. It is implemented in the R package `dynsbm` [41], which we use directly in our experiments. A visual representation of the dynSBM model can be seen in Figure 3.8.

The generative process is as follows:

Let $\{G_t = (V, E_t)\}_{t=1}^{T}$ denote a sequence of graphs on a set of N nodes, with symmetric adjacency matrices $z^{(t)} = [z_{ij}^{(t)}]_{i,j=1}^{N}$ (no self-loops).

- *Markov chain for node memberships:*
  For each node $i = 1, \dots, N$:
  - $Z_i^{(1)} \sim \text{Categorical}(\alpha)$, sample the initial group label of node i at time 1 from the stationary distribution $\alpha = (\alpha_1, \dots, \alpha_p)$.

  For $t = 2, \dots, T$:
  - $Z_i^{(t)} \mid Z_i^{(t-1)} = r \sim \text{Categorical}(\pi_{r,\cdot})$, sample the group label at time t according to the $p \times p$ transition matrix π.

[3]https://github.com/LaosAntreas/dMMSB

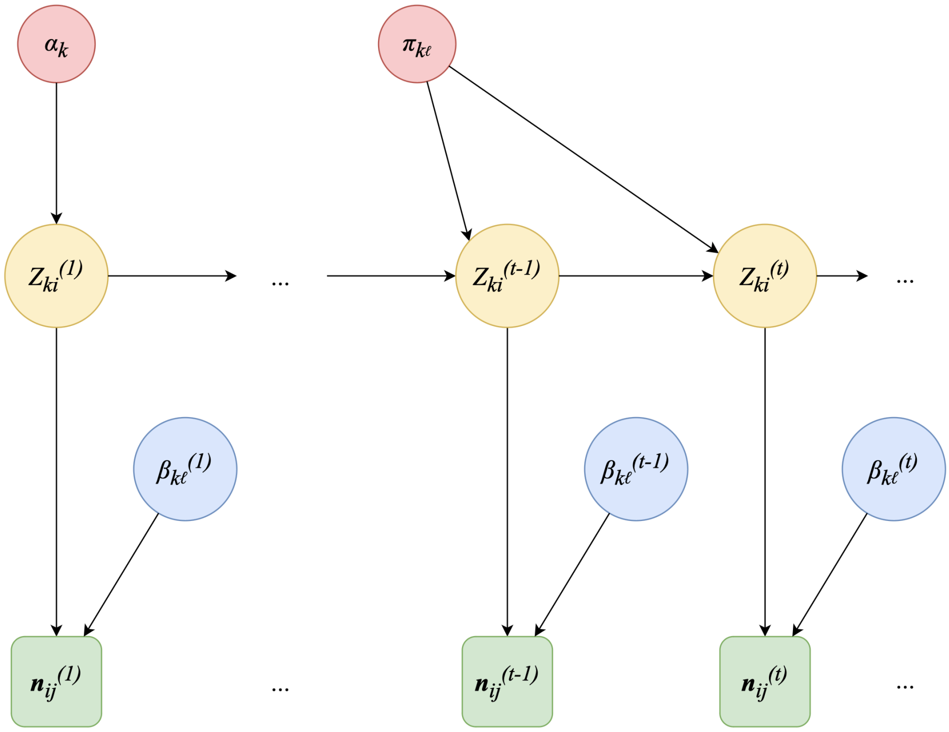

Figure 3.8: Graphical visualization of the dynSBM model.

- *Edge generation given memberships:*
  For each pair of nodes $(i, j) \in [1, N] \times [1, N]$ with $i \neq j$, at each time $t = 1, \ldots, T$:
  - $z_{ij}^{(t)} \mid (Z_i^{(t)} = k,\, Z_j^{(t)} = \ell) \sim \text{Bernoulli}(\beta_{k\ell}^{(t)})$, sample the edge between nodes i and j, where $\beta_{k\ell}^{(t)} \in [0, 1]$ denotes the connection probability between groups k and ℓ at time t.

Note that the authors also propose ways to treat weighted graphs, but none of their approaches includes our Poisson multi-edges, hence we resort to the binary symmetric graphs for our evaluation.

3.5 Results

Having described the data sources, the synthetic protocol, and the baseline models in Section 3.4, we now present the empirical evaluation of dynSNetOC. The aim of this section is threefold. First, we verify that the approximate inference procedure recovers the latent parameters of a graph drawn from the model itself, using the synthetic dataset of Section 3.4.1. Second, we assess whether dynSNetOC produces semantically meaningful and temporally coherent overlapping communities on real-world dynamic networks, using the Reuters Terror corpus and the European Flights network. Third, on both real-world datasets we compare its performance against the dMMSB and dynSBM baselines of Section 3.4.3 to quantify the empirical gains that follow from jointly modeling sparsity, power-law degrees, and overlapping community structure.

Across all three experiments, model fit is assessed primarily through the posterior predictive distribution of the degree sequence: we draw 500 graphs from the fitted posterior at each timestep and report the 95% Posterior Predictive Credible Interval (PPCI) over the empirical degree distribution.

3.5.1 Synthetic Results

We run the model's inference algorithm on the synthetic dataset of Section 3.4.1. We ran the algorithm for 10,000 warmup steps and 10,000 sample steps discarding every other sample for memory conservation.

The sociability weights w_{0i} and community scores $\beta_{ki}^{(t)}$ are well recovered by the posterior. The temporal evolution of the scores for four representative high-degree nodes is shown in Figure 3.9 and the sociability weights of the top 50 highest degree nodes can be seein in Figure 3.10. The credible intervals consistently bracket the true values across all latent parameters, demonstrating that the latent Markov structure as well as the sociability is correctly identified by the sampler. The MCMC trace plots for the hyperparameters $\alpha, \sigma, \tau, \psi, a_1, a_2, b_1$ and b_2 can be found in Appendix.

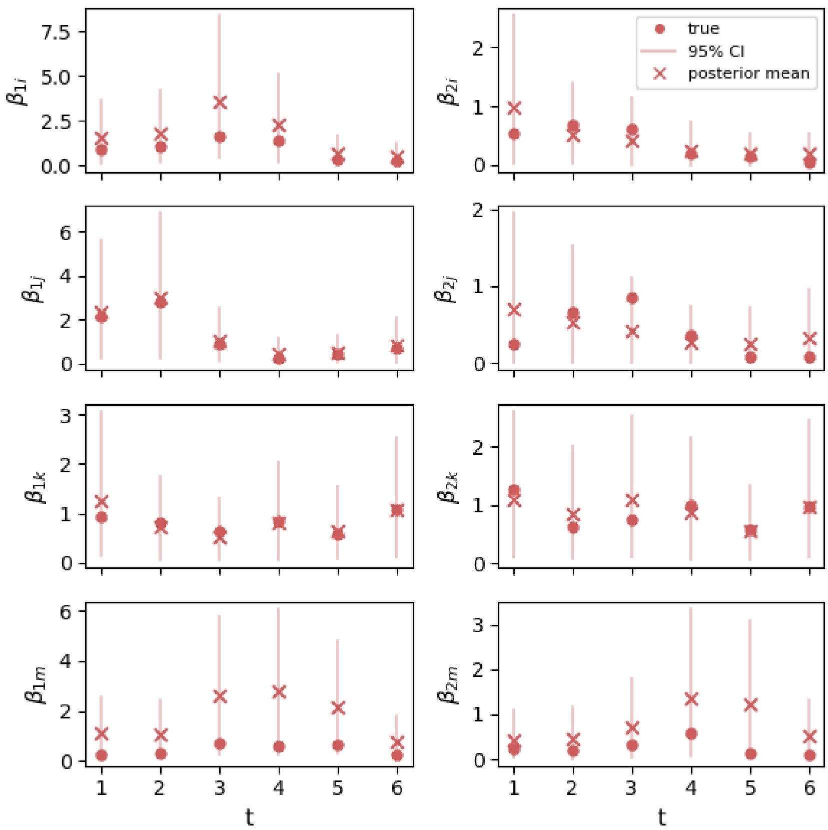

Figure 3.9: Temporal evolution of scores for t=$1, \ldots, 6$ for four high degree nodes (i, j, k, m) simulated from the model. True scores are in dots, and 95% CI in the vertical bars for community 1 (left) and community 2 (right).

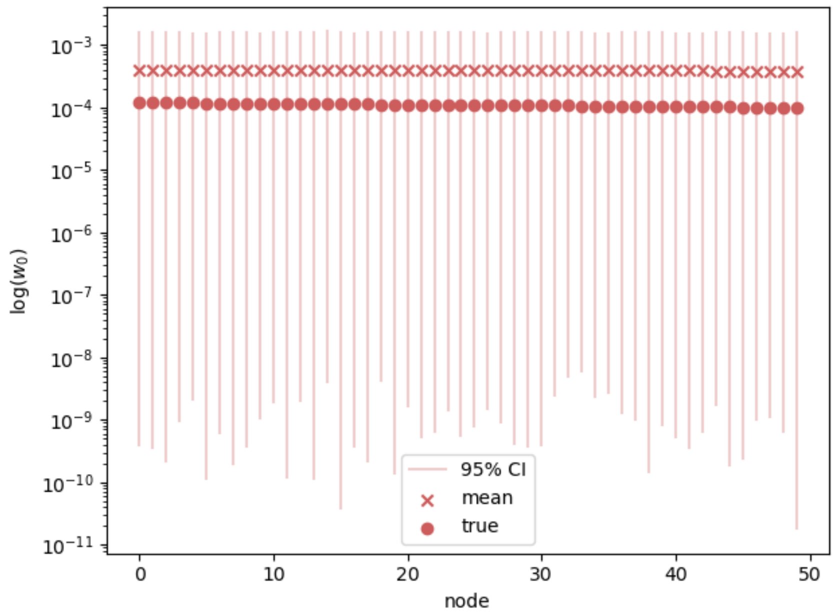

Figure 3.10: w_0 (sociability weights) for the top 50 highest degree nodes in log-scale. True w_0 are in dots, and 95% CI in the vertical bars and posterior mean are in x's.

To assess the posterior predictive fit, 500 graphs are drawn from the fitted posterior at each timestep and the 95% PPCI over the degree sequence is computed. As shown in Figure 3.11, the PPCI envelope covers the empirical degree distribution at every timestep, confirming that dynSNetOC correctly reproduces the power-law, sparse structure of data generated from its own generative process.

Together, the high coverage rates and the accurate degree fit validate that the approximate inference procedure, combined with the reparameterisation optimisations of Section 3.3, is capable of reliably recovering the latent structure of a graph drawn from the dynSNetOC model.

3.5.2 Reuters Terror Results

We apply dynSNetOC to the Reuters Terror word co-occurrence network (Section 3.4.2.1) with $p = 4$ communities, following the work on this corpus [42, 43]. We also tested $p = 5$, but found $p = 4$ to offer better interpretability. With $p = 4$, exactly four community slots are active per timestep, but their thematic content can shift freely, so the effective number of distinct themes across six weeks may exceed p.

Inference was conducted on one NUTS chain of 250,000 samples, 130,000 of which were discarded for warmup. To assess if the inference has converged we evaluate the trace plots of the hyperparameters of the model, which can be seen in Figure II.1 in the Appendix.

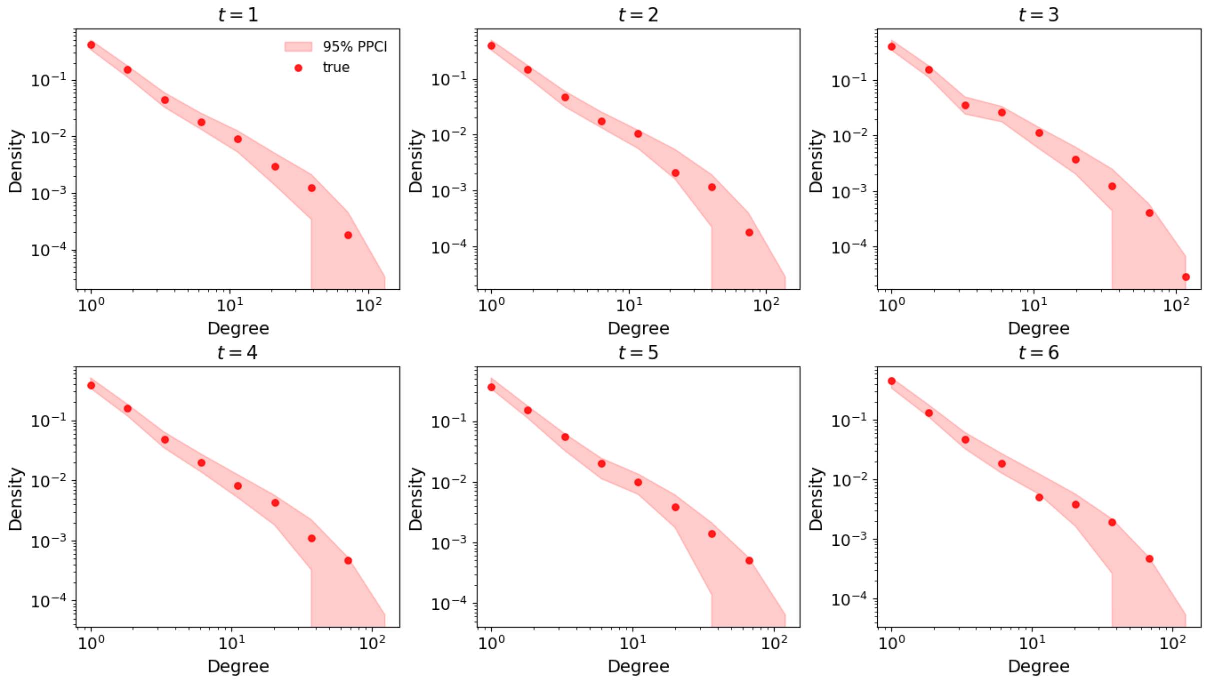

Figure 3.11: Degree distribution in log-log scale for the synthetic dataset for $t = 1, 2, 3$ (top row) and $t = 4, 5, 6$ (bottom row). Empirical degree in red dots, and 95% PPCI in the shaded region.

Community	Representative words
Afghanistan	*Taliban, Al-Qaeda, Bin laden, Afghanistan*
anthrax	*office, post, security, senate, senator, newspaper, sample, Brokaw*
attack	*pentagon, new york, plane, hijack, WTC, attack*
political	*foreign minister, official, military, economy, financial*
security	*passenger, federal, hijacker, airline, agent, FBI, man*

Table 3.1: A high level view of representative words for each community for the 6 weeks.

3.5.2.1 Interpretation of Communities

The model recovers five thematically coherent communities: **attack**, **Afghanistan**, **political**, **anthrax**, and **security** (summarized in Table 3.1). Their temporal evolution is shown in the Sankey diagram of Figure 3.12 and the per-word pie charts of Figure 3.13.

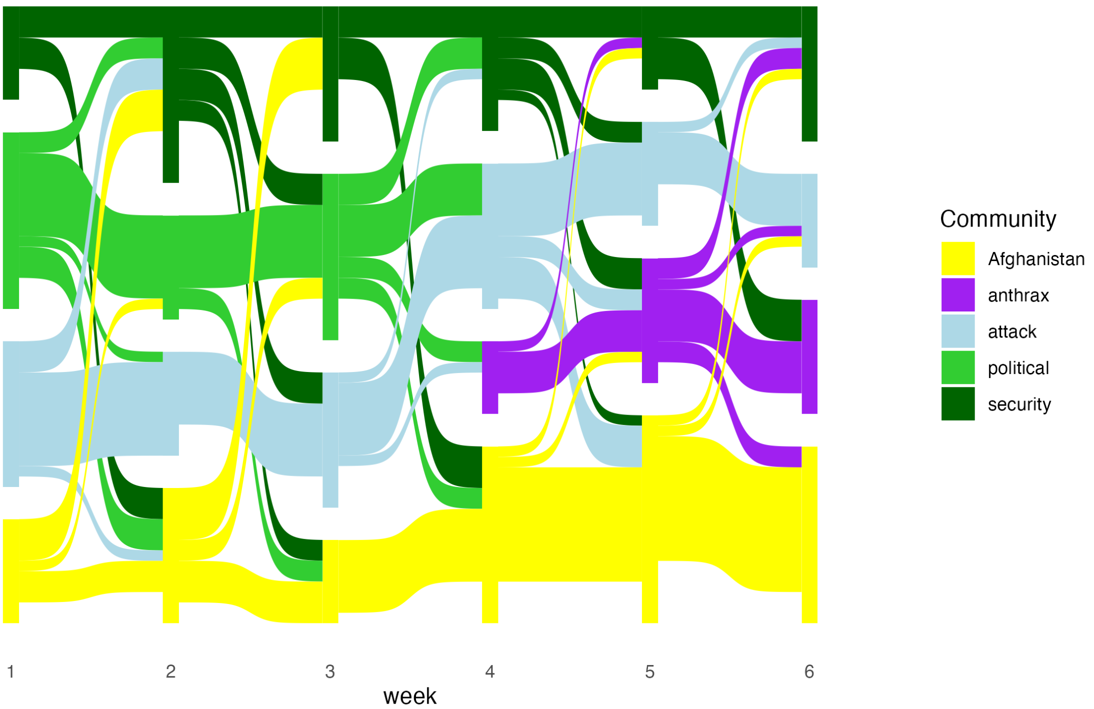

Figure 3.12: Sankey plot on the dynamic behavior of the words memberships for reuters using dynSNetOC. Colors correspond to communities which evolve along the weeks.

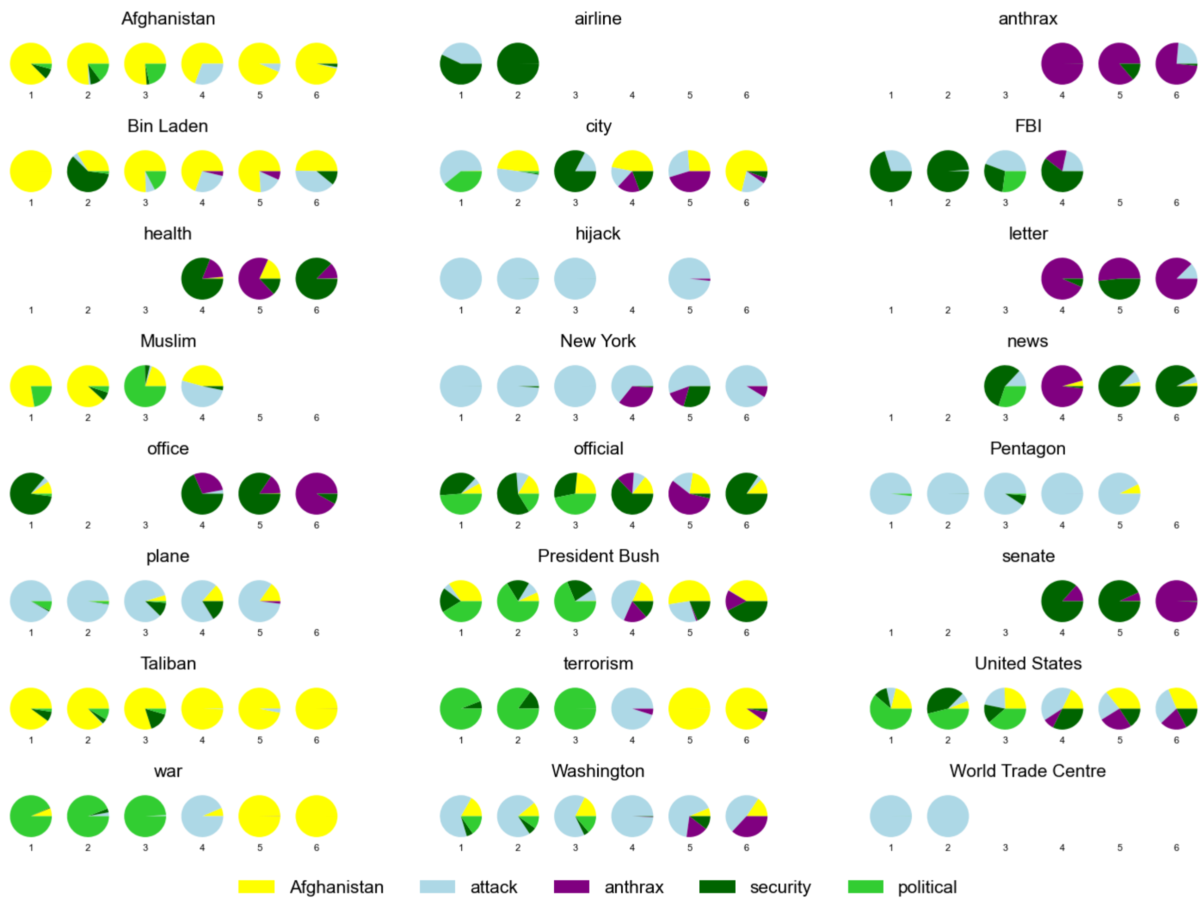

Figure 3.13: Pie charts with the % of community affiliations for some high degree words at each timestep using our model. If a word is not present at a certain timestep (i.e. it is not in the 50 highest degree nodes) then its pie plot doesn't exist.

The temporal trajectory of each community closely mirrors the real-world sequence of events between September and November 2001, validating the semantic coherence of the inferred structure:

- **attack** initially centers on the WTC events (*Pentagon, plane, hijack, WTC*). At week 4, when the US invaded Afghanistan, words from 'political' (*Bush, war, terrorism*) migrate into this community, shifting its meaning toward the broader "War on Terror."
- **Afghanistan** starts with Taliban-related vocabulary (*Bin Laden*, *Muslim*) and expands in weeks 4–5 to include war-in-Afghanistan terminology.
- **political** captures the political and financial response (*foreign minister, economy, financial*) and is prominent in weeks 1–3 before dissolving into **attack** by week 4.
- **anthrax** emerges at week 4, coinciding with the first anthrax-letter death on October 5. Key words include *bacterium*, *spore*, *antibiotic*, *letter*, and notably *Brokaw* (the NBC journalist who received a contaminated letter).

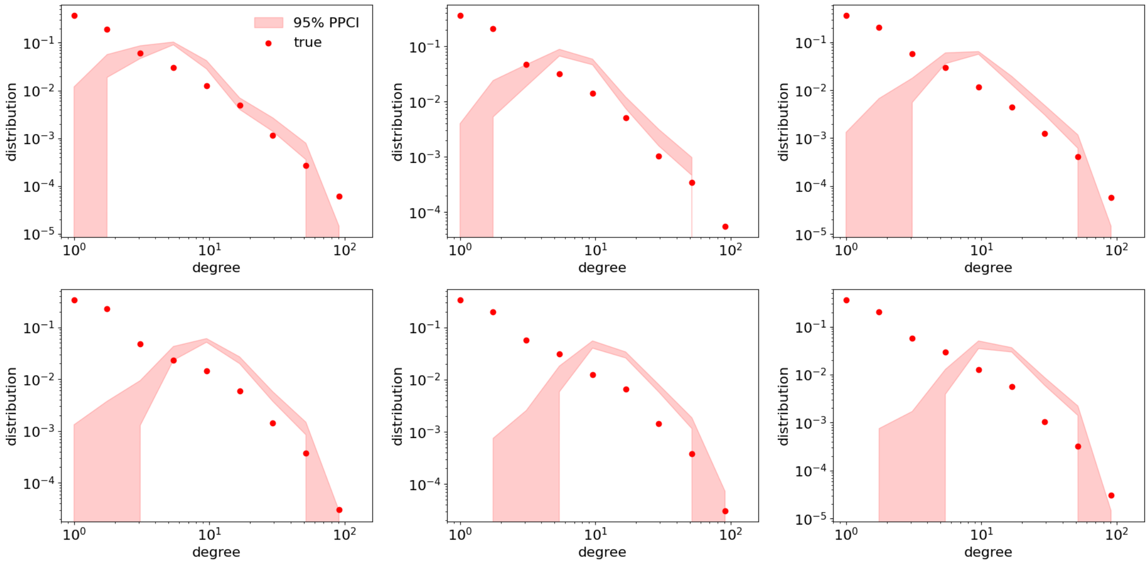

Figure 3.14: Degree distribution of dynSBM for the Reuters network in log-log scale, empirical in red dots and 95% PPCI in shaded region for $t = 1, 2, 3$ (top row), and $t = 4, 5, 6$ (bottom row).

- **security** evolves from airport security (*passenger, airline, FBI*) in early weeks toward national and then health security, overlapping with 'anthrax' in weeks 5–6.

The soft overlapping memberships reveal nuances that hard clustering cannot: *city* shifts from New York to Kandahar; *letter*, *senate*, and *health* straddle **anthrax** and **security**; President Bush travels through **political**, **attack**, and **Afghanistan** in line with the historical arc of events.

3.5.2.2 Baseline Comparisons

We now run the reuters dataset on the two baseline models with $p = 4$ to provide a consistent comparison between the models.

DynSBM: The model produces a poor degree fit as seen in Figure 3.14 and assigns roughly 80% of nodes to a single community at each timestep, yielding no meaningful thematic structure. The failure stems from the model's incompatibility with power-law sparsity.

dMMSB: This model fits the degree distribution better (Figure 3.15), but the block-interaction matrix $\beta \in [0, 1]^{p \times p}$ degenerates to a single non-zero diagonal entry. Community assignment reduces to sorting nodes by degree, providing no semantic information.

Both baselines fail to capture both sparsity and meaningful community structure simultaneously, as expected. This is consistent with the theoretical result of Orbanz & Roy [32], who show that exchangeable graph models are necessarily either dense or empty in the limit, making them fundamentally incompatible with the sparse, power-law networks observed in practice. dynSNetOC escapes this limitation by moving away from the adjacency matrix representation and

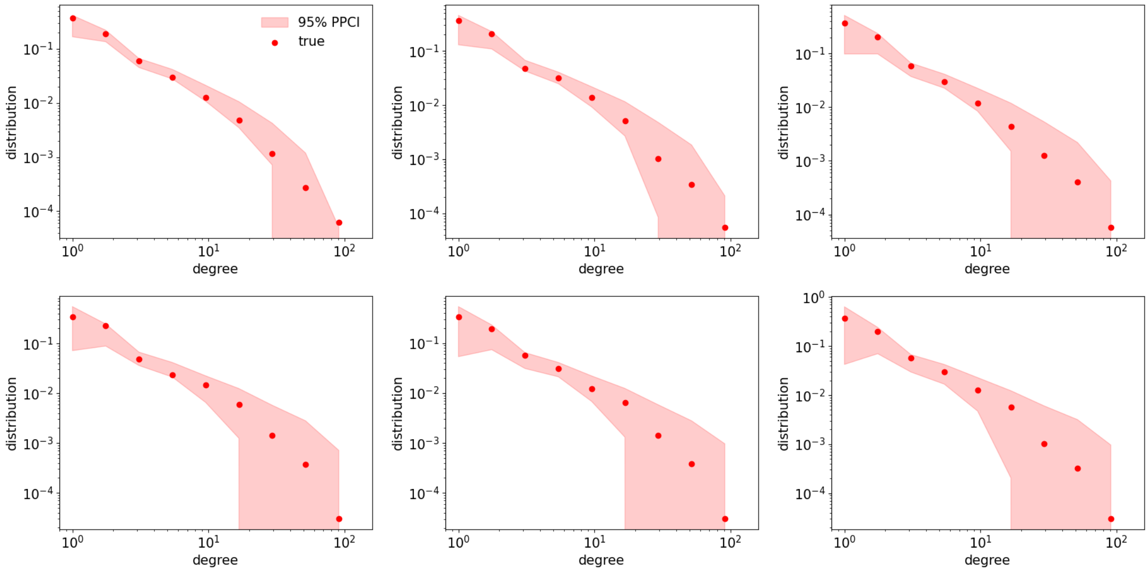

Figure 3.15: Degree distribution in log-log scale of dMMSB for the Reuters network, empirical in red dots and 95% PPCI in shaded region for $t = 1, 2, 3$ (top row), and $t = 4, 5, 6$ (bottom row).

instead building on the exchangeable random measure framework of Caron & Fox [8], which redefines exchangeability at the level of the underlying point process rather than the node labels. The results above confirm that this construction is essential for recovering both a faithful degree fit and interpretable dynamic communities on this dataset.

3.5.3 European Flights Results

We apply dynSNetOC to the European Flights network (Section 3.4.2.2) with $p = 4$. We also tested $p = 3$, and $p = 5$ all consistently providing thematic groupings, but decided to use $p = 4$ as a balance between clarity and fit.

Inference was conducted on one NUTS chain of 10,000 samples and 10,000 warmup steps (warmup and samples were set large enough for the model to converge). To assess convergence, again, we evaluate the traceplots of the MCMC chain, witch can be seen in Figure III.1 in the Appendix. To assess the posterior predictive fit, 500 graphs are drawn from the fitted posterior at each timestep and the 95% PPCI over the empirical degree sequence is computed. As shown in Figure 3.17, the PPCI envelope covers the empirical degree distribution at every timestep, confirming that dynSNetOC correctly reproduces the power-law, sparse structure of the Flights network across all four months.

Community	Representative Airports
Eastern Mediterranean	Istanbul Ataturk Airport (Istanbul, **TR**) Athens International Airport (Attiki, **GR**) Sabiha Gokcen International Airport (Istanbul, **TR**) Ben Gurion Airport (Tel Aviv, **IL**) Bucharest Henri Coandă International Airport (Ilfov, **RO**)
Nordics	Oslo Airport (Akershus, **NO**) Stockholm Arlanda Airport (Stockholms lan, **SE**) Copenhagen Airport (Hovedstaden, **DK**) Helsinki Airport (Uusimaa, **FI**) Bergen Airport (Hordaland, **NO**)
Central Europe	Heathrow Airport (England, **GB**) Paris Charles de Gaulle Airport (Ile-de-France, **FR**) Frankfurt Airport (Hessen, **DE**) Rome–Fiumicino International Airport (Lazio, **IT**) Amsterdam Airport Schiphol (Noord-Holland, **NL**) Munich Airport (Bayern, **DE**)
UK & Spain	Adolfo Suarez Madrid-Barajas Airport (Madrid, **ES**) Palma de Mallorca Airport (Illes Balears, **ES**) Barcelona El Prat Airport (Catalunya, **ES**) Manchester Airport (England, **GB**) Edinburgh Airport (Scotland, **GB**)

Table 3.2: A high level view of representative airports for each community.

3.5.3.1 Interpretation of Communities

The model recovers four geographically coherent communities: **Eastern Mediterranean**, **Nordics**, **Central Europe**, and **UK & Spain** as summarized by representative airports in Table 3.2.

Their spatial layout is visualised on the world and Europe-only maps of Figure 3.16 where each node is coloured by its maximum-affiliation community, and the per-airport pie charts of Figure 3.18 expose the overlapping memberships of a selection of high-degree hubs and regional airports.

The geographic structure of the inferred communities aligns strikingly well with the operational geography of European aviation, confirming that the detected structure is semantically coherent:

- **Eastern Mediterranean** concentrates the airports of Greece, Turkey, Cyprus, Israel, and Romania (Istanbul Ataturk, Athens International, Sabiha Gokcen, Ben Gurion, Bucharest Henri Coandă). It also absorbs Dubai International, reflecting the dense Gulf–Eastern-Mediterranean corridor and the role of Dubai as a transfer hub for the region.
- **Nordics** groups the Scandinavian and Finnish hubs (Oslo, Stockholm Arlanda, Copenhagen, Helsinki, Bergen), capturing the tightly interconnected intra-Nordic flight network

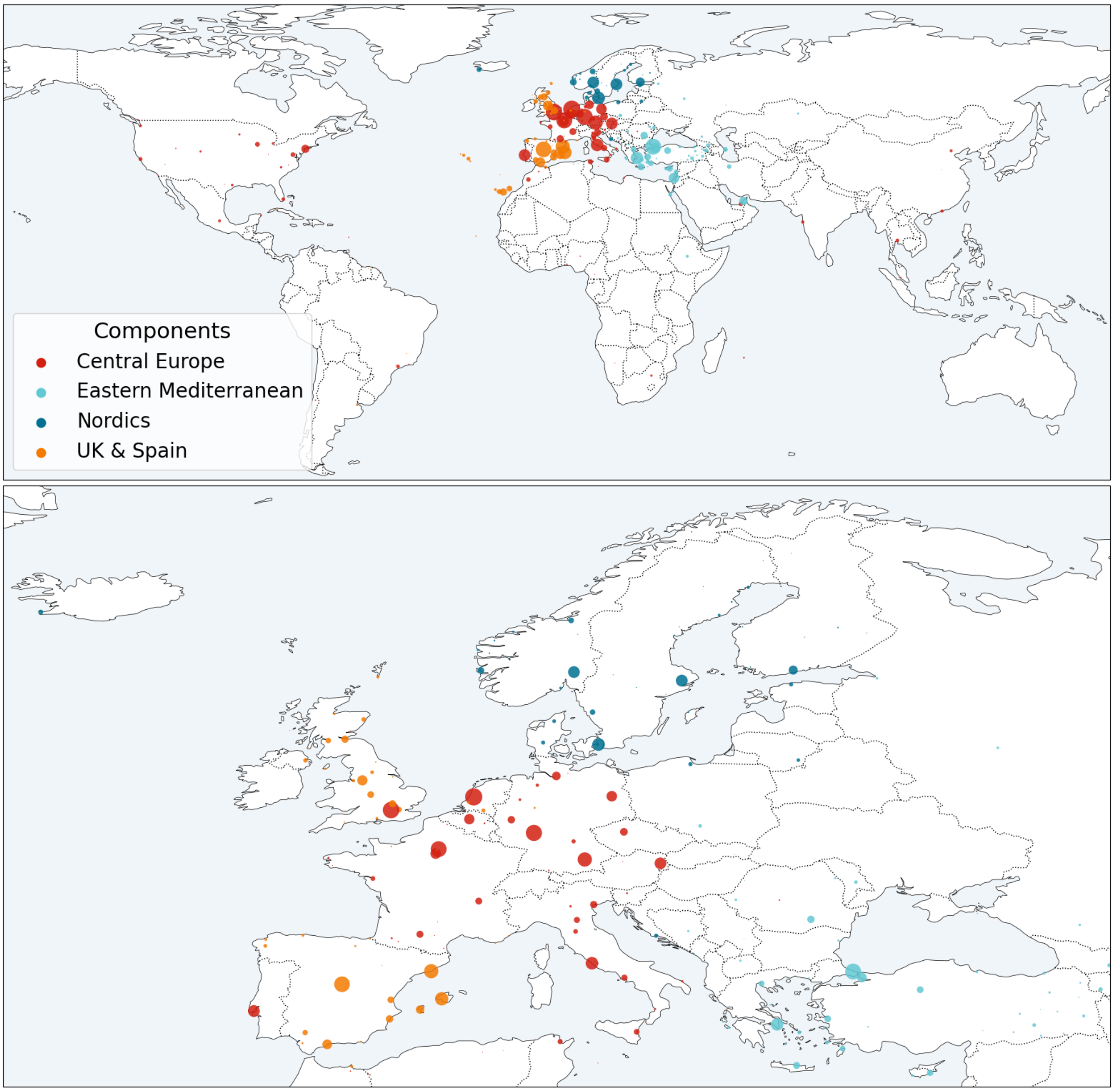

Figure 3.16: Visualization of the community assignments European Flights network for a sample month Map of the earth (Top) and a zoomed in version showing Europe only (Bottom). Each node represents an airport, with node size proportional to its degree and color assigned as the maximum community affiliation for that node.

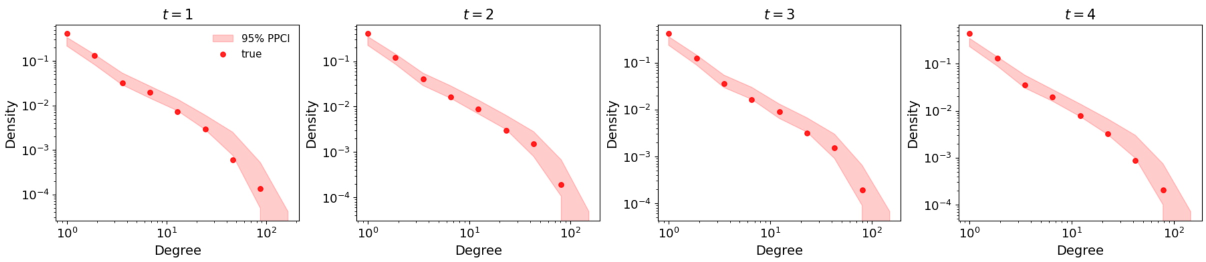

Figure 3.17: Degree distribution of dynSNetOC for the Flights network in log-log scale, empirical in red dots and 95% PPCI in shaded region for t = 1, 2, 3, 4 from left to right

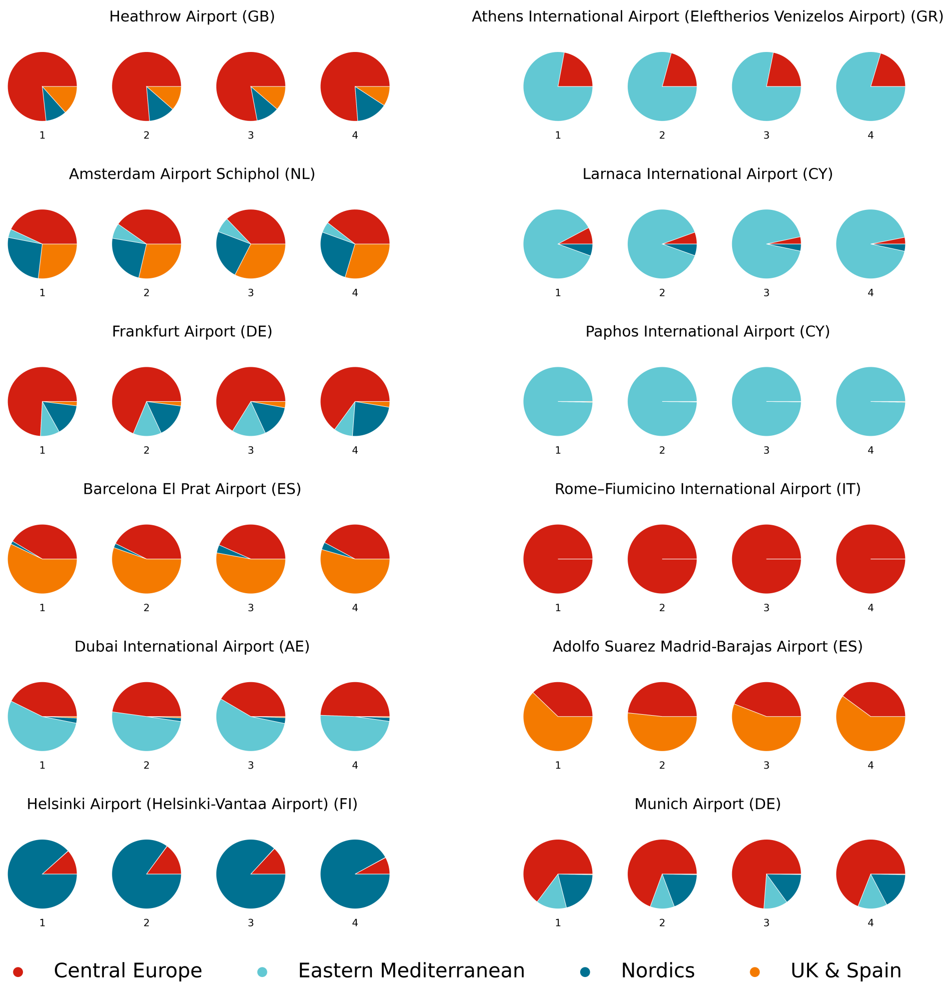

Figure 3.18: Pie charts with the % of community affiliations for some airports at each timestep using dynSnetOC.

that the model isolates from the rest of continental Europe.

- **Central Europe** is the dominant community of the network, gathering the principal hubs of Western and Central continental Europe together with London (Heathrow, Paris Charles de Gaulle, Frankfurt, Amsterdam Schiphol, Munich, Rome–Fiumicino). On the map of Figure 3.16, this community forms the dense red core stretching from the British Isles to Italy.
- **UK & Spain** couples the Iberian airports (Madrid-Barajas, Palma de Mallorca, Barcelona El Prat) with the British secondary hubs (Manchester, Edinburgh), reflecting the dense tourist and low-cost-carrier corridors that link the Iberian peninsula to the United Kingdom outside of the major Central-European hubs.

The soft overlapping memberships of Figure 3.18 reveal nuances that a hard partition cannot. The major continental hubs hold dominant **Central Europe** mass but maintain visible affilia-

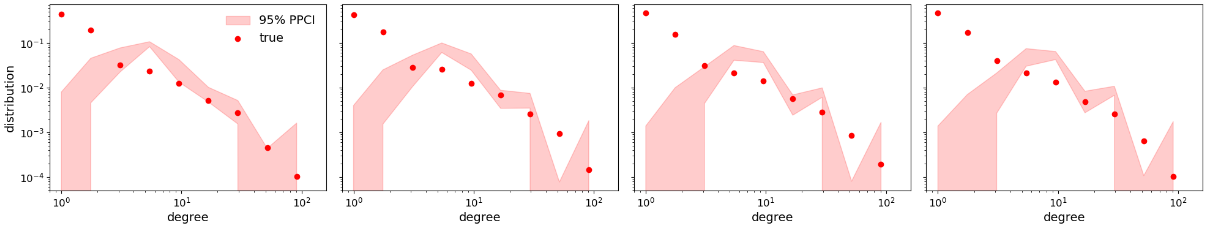

Figure 3.19: Degree distribution of dynSBM for the Flights network in log-log scale, empirical in red dots and 95% PPCI in shaded region for $t = 1, 2, 3, 4$ from left to right.

tions to neighbouring communities: Heathrow, Amsterdam Schiphol, Frankfurt and Munich all pick up components from the other communities, consistent with their role as transfer points for traffic into and out of those regions. Beyond their spatial overlap, the inferred communities evolve smoothly rather than abruptly across the four months. The posterior places the temporal-coupling parameter at $\psi \approx 13$ on average, indicating strong autocorrelation between consecutive timesteps. This largely reflects the binarisation of our graph: although the volume of flights between a given pair of airports varies seasonally, so long as the count does not fall to zero the binary edge persists unchanged, leaving the model little structural signal to attribute to time. In contrast to the rapid weekly turnover observed on the Reuters Terror corpus, this regime confirms that dynSNetOC accommodates both highly dynamic and largely static temporal graphs within the same generative framework.

3.5.3.2 Baseline Comparisons

DynSBM: As with the Reuters Terror network, the model fails to capture both the degree distribution (Figure 3.19) and the community structure of the European Flights network. The fitted solution is entirely degenerate as all nodes belong to a single community at every timestep, rendering the model ineffective as a community detection tool on this dataset.

dMMSB: The model again fails to capture the community structure of the dataset, with the β matrix having only a single nonzero entry on the diagonal, which essentially captures node sociability. This is evident in Figure 3.20 which somewhat captures the degree distribution but fails to provide any meaningful community assignment.

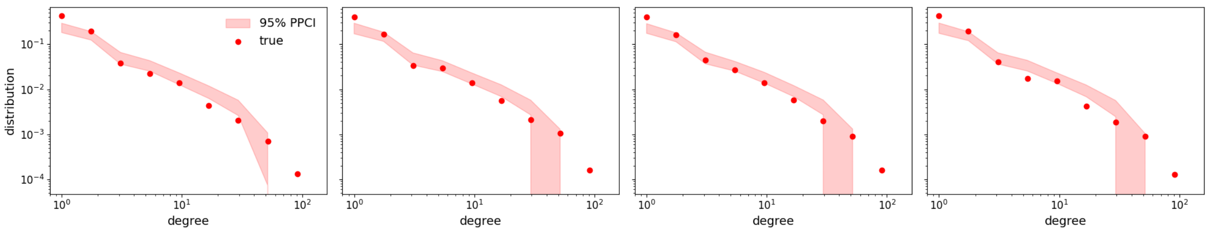

Figure 3.20: Degree distribution of dMMSB for the Flights network in log-log scale, empirical in red dots and 95% PPCI in shaded region for $t = 1, 2, 3, 4$ from left to right.

Both baselines confirm, on a network of qualitatively very different nature from the Reuters Terror corpus, the same pattern already observed there: classical SBM-based approaches cannot simultaneously accommodate sparsity, power-law degree distributions, and overlapping community structure, while dynSNetOC delivers both a faithful degree fit and an interpretable set of dynamic overlapping communities on this dataset as well.

Taken together, the synthetic and real-world experiments confirm that dynSNetOC achieves what it was designed to do: it recovers sparse, power-law degree structure and interpretable, temporally consistent overlapping communities on networks where dynamic blockmodel baselines fail entirely. These strengths, however, come at a computational price, MCMC inference over a high-dimensional latent space takes time to run. The next chapter takes up exactly these limitations, asking whether a graph neural network can recover comparable community structure under the same Bernoulli–Poisson observation model while scaling on GPU hardware.

4 Deep Learning-based Network Modeling

TABLE OF CONTENTS

The Bayesian model of Chapter 3 provides a principled and interpretable account of dynamic overlapping community structure: it captures sparsity through the Caron–Fox point-process construction, models overlap through per-node sociability vectors, and couples successive timesteps through a latent Markov process. These strengths come at a cost. Inference is performed by MCMC over a high-dimensional continuous parameter space, and although we exploit the sparsity of the Poisson likelihood and use a non-centered reparameterization to improve the posterior geometry, the type of computational workload is not efficiently parallelizable at scale.

This chapter explores a second approach to the same problem, one that is scalable and compatible with modern GPU hardware. Rather than treating the community affiliations as latent variables to be inferred, we generate them with a temporal graph neural network and perform Maximum Likelihood Estimation (MLE) with stochastic gradient descent. The observation model is unchanged: given non-negative affiliations $\mathbf{F}^{(t)} \in \mathbb{R}_{\geq 0}^{L \times p}$ for each timestep (equivalent to $w_{ik}^{(t)}$ in dynSNetOC), edges are sampled from a Bernoulli–Poisson likelihood whose rate is the inner product of affiliation vectors. This chapter's contribution is the temporal extension of the NOCD model of Shchur & Günnemann [10].

The chapter is organized as follows. Section 4.1 describes the necessary background information for the rest of the chapter, Section 4.2 outlines the deep learning architecture used to tackle the task, and Section 4.3 evaluates the proposed architecture on the datasets described in the previous chapter.

4.1 Background

This section provides the technical background specific to this chapter. Chapter 2 introduced GNNs and the NOCD model at a high level as part of the related work survey. Here we describe the components needed to understand the temporal extension: the Bernoulli-Poisson generative model underlying NOCD, the GCN architecture used to parameterize the affiliations, the training objective and its scalable stochastic approximation, and the class of temporal GNNs that supply the dynamic encoder.

4.1.1 Graph Convolutional Networks

Graph Convolutional Networks [44] are a type of Message Passing Neural Network (see Definition 19) that generalize the idea of convolution from regular grids (images) to graphs. Each layer aggregates the features of a node's immediate neighbors, applies a shared linear transformation, and passes the result through a non-linearity. Stacking L layers extends the receptive field to the L-hop neighborhood.

The layer-wise propagation rule is:

$$\mathbf{H}^{(l+1)} = \sigma\left(\hat{\mathbf{A}}\,\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right), \tag{4.1}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{N\times d_l}$ is the matrix of node representations at layer l, $\mathbf{W}^{(l)}$ is a learnable weight matrix, σ is a pointwise non-linearity (ReLU), and

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}, \qquad \tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}, \qquad \tilde{D}_{ii} = \textstyle\sum_j \tilde{A}_{ij}$$

is the symmetrically normalised adjacency matrix with added self-loops. The self-loops ensure each node includes its own representation when aggregating; the symmetric normalisation prevents gradient instability during training.

4.1.2 Neural Overlapping Community Detection

The Neural Overlapping Community Detection (NOCD) model [10] combined the Bernoulli-Poisson observation model with a graph convolutional network encoder.

Bernoulli-Poisson (BerPo) observation model. Given a non-negative community affiliation matrix $\mathbf{F} \in \mathbb{R}_{\geq 0}^{L\times p}$ the BerPo model generates binary adjacency matrix entries independently as

$$z_{ij} \sim \text{Bernoulli}(1 - \exp(-\mathbf{F}_i\mathbf{F}_j^\top)) \tag{4.2}$$

where $\mathbf{F}_u$ is the row vector of community affiliations of node u. . This is precisely the observation model that appears throughout this thesis. As noted in Section 2.2.4, the Caron–Fox [8] edge probability that can be seen in Equation 2.2, $Pr(z_{ij} = 1|w_i, w_j) = 1 - e^{-2w_iw_j}$, is a univariate special case of Equation 4.2.

The negative log-likelihood of the BerPo model is

$$-\log(n|\mathbf{F}) = -\sum_{(i,j)\in E} \log(1 - \exp(-\mathbf{F}_i\mathbf{F}_j^\top)) + \sum_{(i,j)\notin E} \mathbf{F}_i\mathbf{F}_j^\top \tag{4.3}$$

The graphs that we are evaluating these models on are sparse. This leads the second term of Equation 4.3 to dominate the loss. Shchur & Günnemann [10] counteract this by balancing edge and non-edge contributions equally:

$$\mathcal{L}_{\text{BP}}(\mathbf{F}) = -\mathbb{E}_{(i,j)\sim P_E}[\log(1 - \exp(-\mathbf{F}_i\mathbf{F}_j^\top))] + \mathbb{E}_{(i,j)\sim P_N}[\mathbf{F}_i\mathbf{F}_j^\top] \tag{4.4}$$

where P_E and P_N are uniform distributions over edges and non-edges respectively.

GCN encoder. Instead of treating **F** as a free variable, NOCD generates it from the observed graph structure and (optionally) node features via a two-layer Graph Convolutional Network [44].

$$\mathbf{F} \;=\; \text{GCN}_\theta(A, X) \;=\; \text{ReLU}\Big(\hat{A}\,\text{ReLU}(\hat{A}XW^{(1)})W^{(2)}\Big)\,, \tag{4.5}$$

where $\hat{A}$ is a symmetrically normalised adjacency with self-loops, and $W^{(1)}$, $W^{(2)}$ are learned weight matrices. The element-wise ReLU in the final layer enforces the non-negativity requirement of the BerPo model.The MLE objective is then:

$$\theta^\star \;=\; \arg\min_\theta \; \mathcal{L}_{\text{BP}}(\text{GCN}_\theta(A, X))\,. \tag{4.6}$$

4.2 Model Definition

The goal of this part of the thesis is to extend the static NOCD model to accommodate temporal graphs. We refer to the resulting model as dNOCD (dynamic NOCD), an overview of which is given in Figure 4.1.

It is worth noting that several alternative architectures were explored during development, including recurrent mechanisms and attention-based cross-time aggregation. However, upon evaluation across both datasets, the simpler stateless encoder with the smoothness penalty produced communities of superior coherence, and was therefore preferred.

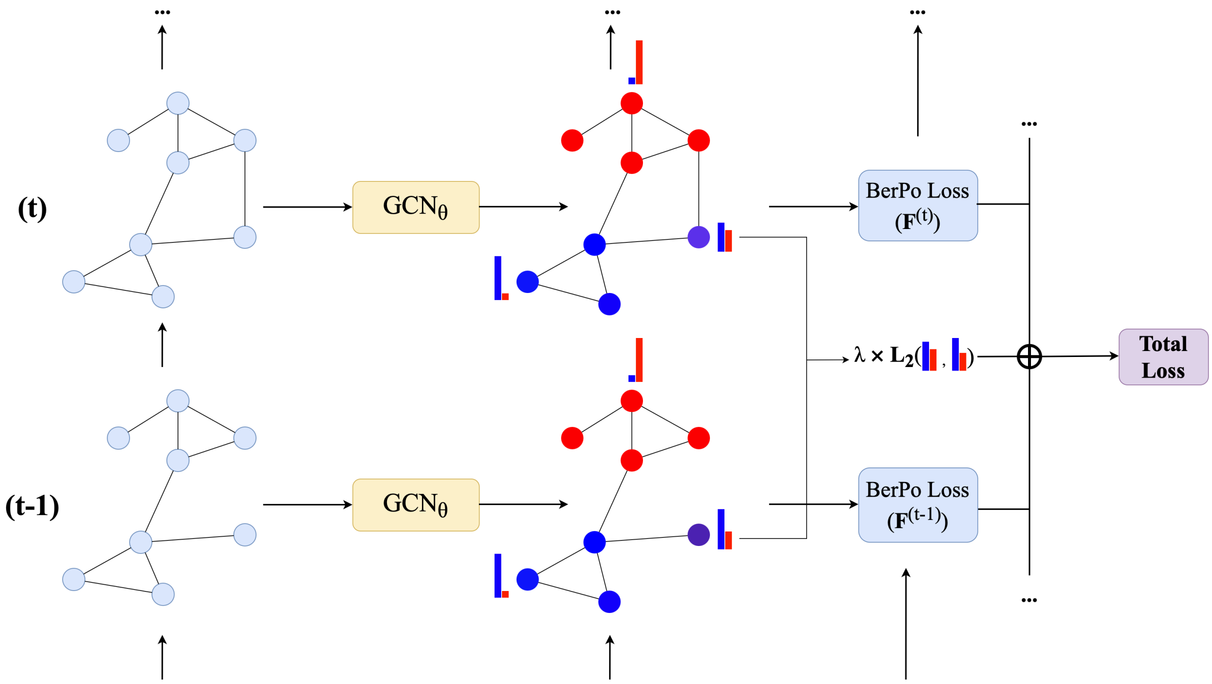

Figure 4.1: Visual representation of the dNOCD model and loss function for timesteps $t-1$ and t.

4.2.1 Architecture

A temporal graph is observed as a sequence of adjacency matrices $\{A^{(t)}\}_{t=1}^{T}$ on a common node set of size L, together with (optional) per-snapshot node feature matrices $\{X^{(t)}\}_{t=1}^{T}$. Our goal is to produce, for every timestep t, a non-negative affiliation matrix $\mathbf{F}^{(t)} \in \mathbb{R}_{\geq 0}^{L\times p}$ that obeys the BerPo observation model (Eq. 4.2) on the corresponding snapshot. The simplest way to lift the static NOCD encoder of Equation 4.5 to the dynamic setting is to apply it independently at every timestep with shared parameters. Concretely,

$$\mathbf{F}^{(t)} \;=\; \mathrm{GCN}_{\theta}\big(A^{(t)}, X^{(t)}\big)\,, \qquad t = 1, \ldots, T. \tag{4.7}$$

We write $\mathcal{F} = (\mathbf{F}^{(1)}, \ldots, \mathbf{F}^{(T)})$ for the full sequence of affiliation matrices produced by the encoder. Following the original NOCD paper, we apply batch normalisation and dropout after the first graph convolution layer and add ℓ_2 regularization to all weight matrices.

4.2.2 Training Objective

The objective combines a per-snapshot reconstruction loss and a temporal smoothness penalty.

4.2.2.1 Reconstruction

The BerPo loss of Eq. 4.3 is the negative log-likelihood of a single observed adjacency matrix A under the affiliations **F**. To extend it to a sequence $\{A^{(t)}\}_{t=1}^{T}$ we assume that, conditional on $\mathcal{F}$, the snapshots are independent. Under this assumption the joint likelihood factorises across time,

$$p\big(\{A^{(t)}\}_{t=1}^{T} \,\big|\, \mathcal{F}\big) \;=\; \prod_{t=1}^{T} p\big(A^{(t)} \,\big|\, \mathbf{F}^{(t)}\big)\,, \tag{4.8}$$

and the joint negative log-likelihood is therefore the sum of the per-snapshot BerPo losses,

$$-\log p\big(\{A^{(t)}\}_{t=1}^{T} \,\big|\, \mathcal{F}\big) \;=\; \sum_{t=1}^{T} \mathcal{L}_{\text{BP}}\big(\mathbf{F}^{(t)}\big)\,. \tag{4.9}$$

For every snapshot we apply the BerPo loss of Eq. 4.3 (or its balanced variant, Eq. 4.4), averaged across time:

$$\mathcal{L}_{\text{recon}}(\mathcal{F}) \;=\; \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_{\text{BP}}\big(\mathbf{F}^{(t)}\big)\,. \tag{4.10}$$

4.2.2.2 Temporal Smoothness

Because the encoder is memoryless, nothing in the architecture prevents the affiliations $\mathbf{F}^{(t)}$ and $\mathbf{F}^{(t+1)}$ from being arbitrarily different even when $A^{(t)}$ and $A^{(t+1)}$ are nearly identical. We discourage such jitter with a quadratic penalty on the change of consecutive affiliation matrices,

$$\mathcal{L}_{\text{smooth}}(\mathcal{F}) \;=\; \frac{1}{T-1}\sum_{t=2}^{T} \frac{1}{Lp}\,\big\|\mathbf{F}^{(t)} - \mathbf{F}^{(t-1)}\big\|^2, \tag{4.11}$$

where $\|\cdot\|^2$ denotes the squared Frobenius norm. Pulling $\mathbf{F}^{(t)}$ towards $\mathbf{F}^{(t-1)}$ amounts to a soft Markov prior on the affiliations and directly mirrors the modelling assumption made in the generative dynSNetOC model of Chapter 3, where the latent scores at t and $t-1$ are tied through a Gamma transition kernel. The strength of the prior is controlled by a single scalar $\lambda_{\text{smooth}} \geq 0$, analogous to the ψ hyperparameter in dynSNetOC.

4.2.2.3 Full Objective

The parameters are obtained by minimising

$$\theta^\star = \arg\min_{\theta} \Big(\mathcal{L}_{\text{recon}}(\mathcal{F}) + \lambda_{\text{smooth}}\, \mathcal{L}_{\text{smooth}}(\mathcal{F}) \Big), \tag{4.12}$$

where $\mathcal{F}$ depends on θ through the encoder of Eq. 4.7.

It is worth clarifying in what sense dNOCD is temporal. The encoder of Equation 4.5 applies the same GCN independently to each snapshot given the adjacency matrix $A_{ij}^{(t)}$, it produces $\mathbf{F}^{(t)}$ without any direct access to the representations computed at adjacent timesteps. Temporality therefore enters the model through the training objective rather than through the architecture: the smoothness penalty of Equation 4.11 penalises large changes between timesteps, encouraging consecutive affiliation matrices to remain close and producing temporally coherent communities in practice. The model is therefor fully parallelizable, stateless and easy to train with standard gradient-based optimization.

4.2.2.4 Optimisation

We minimise Eq. 4.12 with AdamW (decoupled weight decay), clipping the global gradient norm at 5 for stability. Every K epochs the reconstruction loss is re-evaluated on the full sequence and used as a validation signal: the best-so-far parameters are checkpointed and training stops once the validation loss has failed to improve for a fixed number of evaluation rounds. Although the model is trained in a transductive, fully unsupervised fashion, this early-stopping rule yields the same model-selection behaviour as in the static case, and in practice rules out the regime in which the BerPo likelihood collapses onto trivial all-zero affiliations.

4.2.3 Relationship to DynSNetOC

. Table 4.1 summarizes the key differences between dynSNetOC (Chapter 3) and dNOCD. Both models share the BerPo observation model and the goal of recovering dynamic overlapping community affiliations. They differ in how those affiliations are generated and how inference is performed.

4.3 Datasets

We evaluate dNOCD on the same two real-world datasets used to evaluate dynSNetOC in Chapter 3: the Reuters Terror word co-occurrence network (Section 3.4.2.1) and the European Flights network (Section 3.4.2.2). Using identical datasets across both chapters allows for a direct

Property	dynSNetOC	dNOCD
Affiliation representation	Latent Gamma variables $w_{ik}^{(t)}$	GNN outputs $F_{ik}^{(t)}$
Temporal coupling	Latent Markov process (Gamma state-space)	via $\mathcal{L}_{\text{smooth}}$
Observation model	Bernoulli-Poisson (via Poisson multi-edges)	Bernoulli-Poisson (binary)
Inference method	NUTS (MCMC)	MLE via AdamW
Uncertainty quantification	Full posterior	Point estimate only
Scalability	$\mathcal{O}(\text{nnz} \cdot p)$ per MCMC step	$\mathcal{O}(S)$ per gradient step, GPU-parallelizable
Degree heterogeneity	Explicit (w_{0i} base weight)	Not captured

Table 4.1: Comparison of dynSNetOC and dNOCD.

comparison between the two approaches under matched experimental conditions. We refer the reader to Section 3.4.2 for full descriptions of the data sources, preprocessing pipelines, temporal granularity, and structural statistics (sparsity, degree distributions, and active-node counts per timestep).

Both datasets exhibit the properties that motivate this work: sparsity, power-law degree behavior, and overlapping community structure that evolves over time. Therefore, they provide a fair stress test of dNOCD's ability to recover meaningful dynamic affiliations without the explicit nonparametric prior used by dynSNetOC.

4.4 Experiments

For both datasets we use the SmoothGCN variant of dNOCD: a two-layer graph convolutional encoder with hidden dimension 128, ReLU non-linearities, batch normalisation after the first layer, and dropout with retention probability 0.5. The encoder is shared across timesteps; temporal coherence is enforced by the L_2 smoothness penalty of Eq. 4.11 with $\lambda = 0.1$. The input features are the row-normalised adjacency matrix at each timestep, following the attribute-free variant NOCD-G of Shchur and Günnemann [10]. We train with AdamW (learning rate 3×10^{-4}, weight decay 10^{-2}), global gradient-norm clipping at 5, and the balanced BerPo loss of Eq 4.3 to compensate for the strong edge–non-edge imbalance induced by sparsity.

4.4.1 Reuters Terror Results

We apply dNOCD to the Reuters Terror co-occurrence network with $p = 4$ communities, mirroring the choice made for dynSNetOC. The recovered affiliations are summarised by the Sankey diagram of Figure 4.3. which tracks the community membership of the fifty highest-degree words across the six weeks following the September 11 attack, and by the per-word pie charts of Figure 4.2, which expose the soft overlapping structure that the hard partition of the Sankey hides.

Comparison with DynSNetOC (Figures 3.12 and 3.13): The agreement between the two paradigms on this dataset is substantial but not complete. Both models locate the same temporal landmarks: the WTC-centred **attack** community in weeks 1–3, the emergence of **anthrax** at week 4, the migration of war-related vocabulary into **Afghanistan** in weeks 4–5, and the dual **security**/**anthrax** affiliation of *letter*, *office*, *health*, and *senate* in weeks 5–6. The qualitative trajectory of historically diagnostic words (*President Bush, war, city, terrorism*) is also preserved across paradigms. There are, however, two noteworthy discrepancies.

First, dNOCD does not recover a similar political community to dynSNetOC. Under dynSNetOC, **political** is active in weeks 1–3 (carrying *foreign minister, economy, financial, official*) and dissolves into attack by week 4 when anthrax takes its slot in the four-community budget. Under dNOCD, those words are split between **security** and **Afghanistan** from week 1, and the **political** slot never materialises as a coherent community. The Sankey plot of Figure 4.3 reflects this: where Figure 3.12 shows a substantial **political** band in weeks 1–3 that dissipates at week 4, Figure 4.3 shows only a thin light-green flow that is rapidly absorbed into the neighbouring communities. We attribute this difference to the smoothing inductive bias of the graph convolution: a GCN layer averages each node's representation with that of its one-hop neighbours, and **political** vocabulary co-occurs heavily with both **security** (institutional terms) and **Afghanistan** (geopolitical terms), so the encoder has limited incentive to keep it in a separate slot.

Second, the dNOCD Sankey is visibly 'cleaner' than its dynSNetOC counterpart in the sense that flows between communities are thicker and less fragmented, with fewer cross-community transitions. This is partly a property of point estimation: under dynSNetOC, the Sankey is built from posterior-mean affiliations and inherits some of the uncertainty in low-mass assignments, whereas under dNOCD each word receives a single deterministic affiliation vector. It is also partly an effect of the L_2 smoothness penalty, which directly penalises week-on-week changes in $\mathbf{F}^{(t)}$ and therefore favours stable assignments; the latent-Markov coupling of dynSNetOC penalises changes more softly through the Gamma transition kernel. The practical implication is that dNOCD is well-suited to recovering the backbone of the community structure but is less expressive about the gradual semantic drift that characterizes this corpus.

4.4.2 European Flights Results

We apply dNOCD to the European Flights network with $p = 4$, again matching the configuration used for dynSNetOC. The community structure recovered by dNOCD on a sample month is show in Figure 4.4. The per-airport pie charts, as seen in Figure 4.5, expose the overlapping

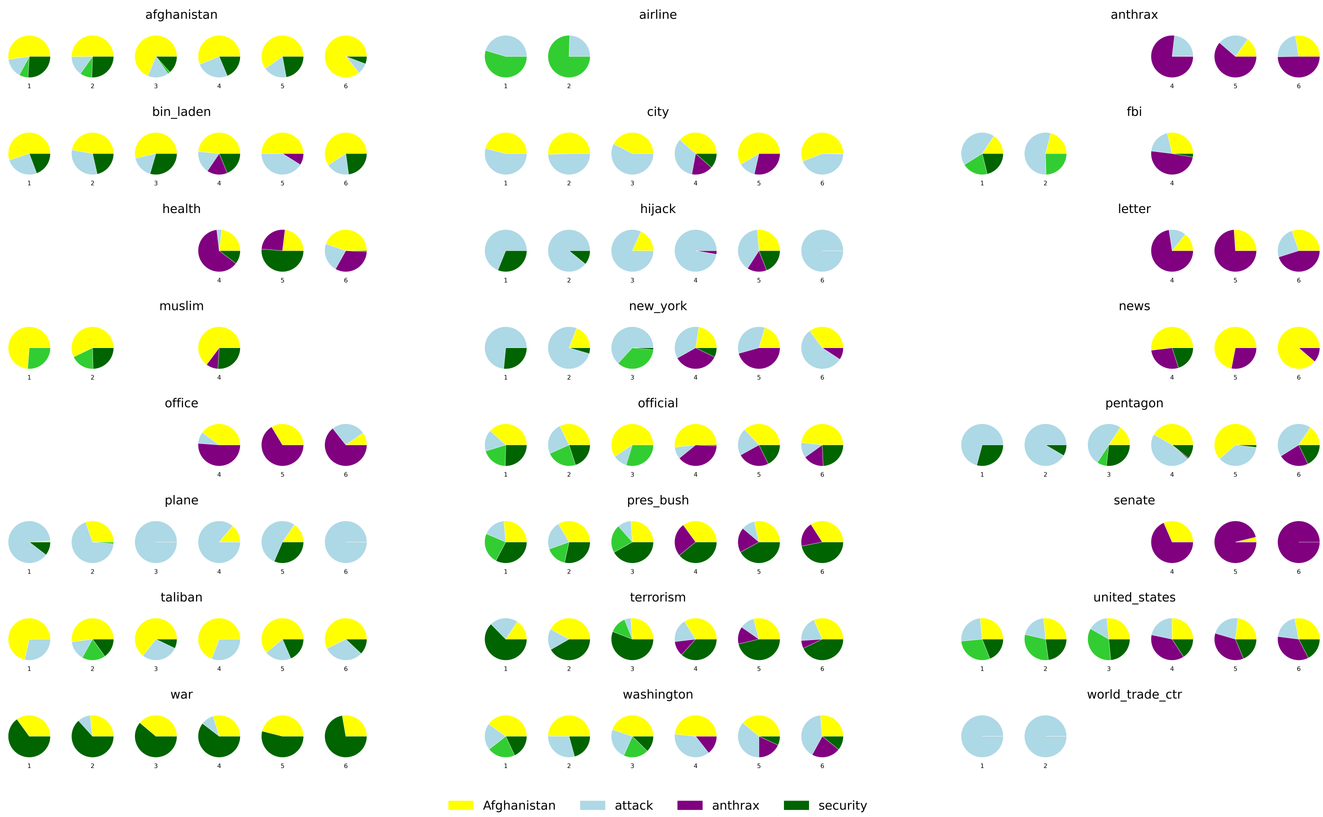

Figure 4.2: Pie charts with the % of community affiliations from dNOCD for some high degree words at each timestep using our model. If a word is not present at a certain timestep (i.e. it is not in the 50 highest degree nodes) then its pie plot doesn't exist.

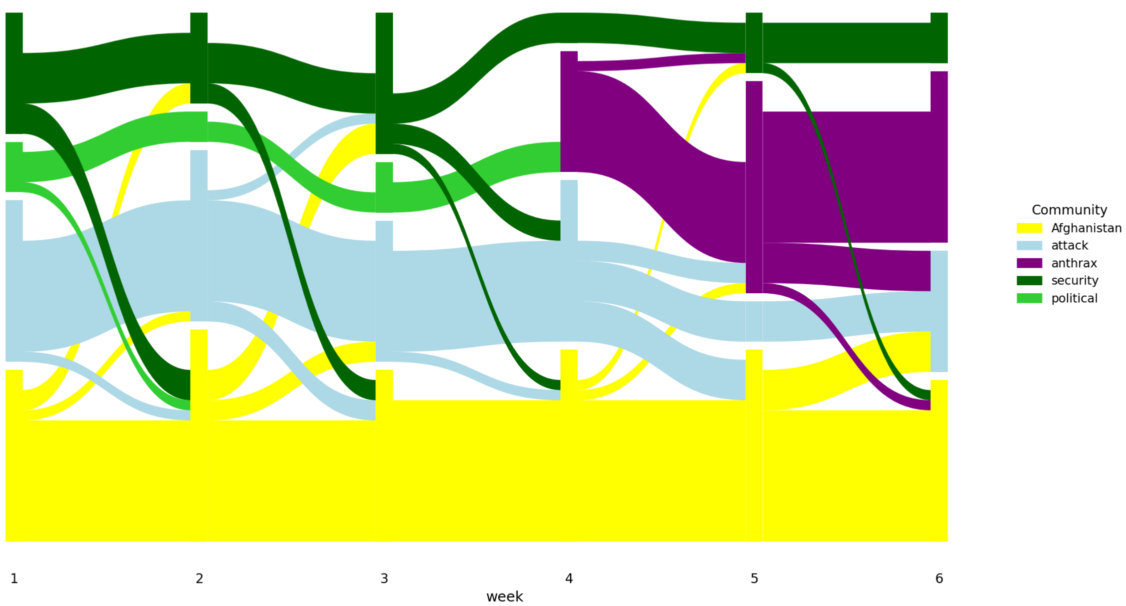

Figure 4.3: Sankey plot on the dynamic behavior of the words memberships for reuters using dNOCD. Colors correspond to communities which evolve along the weeks.

memberships for a selection of major hubs and regional airports.

The model recovers exactly the same four geographically coherent communities as dynSNetOC: **Eastern Mediterranean**, **Nordics**, **Central Europe**, and **UK & Spain**. The spatial layout of Figure 4.4 maps onto the operational geography of European aviation in the same way as Figure 3.16

Comparison with DynSNetOC (Figures 3.18 and 3.16): The agreement between the two paradigms on this dataset is essentially complete: both models recover the same four communities, with the same set of representative airports, and the soft overlapping structure of the major hubs is qualitatively indistinguishable between Figures 3.18 and 4.5. The only systematic difference we observe is that the dNOCD pie charts are slightly flatter (meaning that the distribution of the pie charts changes less over time). This mirrors a remark already made in Section 3.5.3 the empirical Flights network is itself a largely static graph under binarisation, because the binary edge between two airports persists as long as the underlying flight volume is non-zero. dynSNetOC reflects this by placing the temporal-coupling hyperparameter at $\psi \approx 13$ on average; dNOCD reflects it by converging to a regime in which the smoothness penalty dominates, and consecutive affiliation matrices $\mathbf{F}^{(t)}$ and $\mathbf{F}^{(t+1)}$ are nearly identical.

4.4.3 Degree Distribution Fit

While the community structure recovered by dNOCD on both datasets is interpretable and broadly consistent with that of dynSNetOC, the model's ability to reproduce the empirical degree distribution is substantially weaker than what was achieved by the Bayesian nonparametric construction of Chapter 3. Figure 4.6 plots, for both the Reuters Terror and the European Flights networks, the empirical degree distribution against the expected degree implied by the fitted affiliations under the BerPo observation model.

In both cases the heavy-tailed power-law shape that dynSNetOC reproduces faithfully under the posterior predictive (Figures 3.17 and 3.4) is visibly distorted under dNOCD. We attribute this discrepancy primarily to the balanced form of the Bernoulli–Poisson loss used during training (Eq. 4.4). The unbalanced BerPo negative log-likelihood of Equation 4.3 is, by construction, the correct objective for maximum likelihood estimation of $\mathbf{F}$ under the generative model. On a sparse network, however, it is dominated by the non edge contributions against the edge contributions. The balanced variant compensates by reweighting the two sums to have equal aggregate scale, which replaces the dominant non-edge term with one that, per non-edge pair, is orders of magnitude smaller than the per-edge term. This is what makes the model trainable in practice and is the convention adopted by NOCD [10], but it also means that the optimized $\mathbf{F}$ no longer captures degree heterogeneity as it prioritises recovering the structure of the edge set at the expense of correctly modelling the marginal degree sequence.

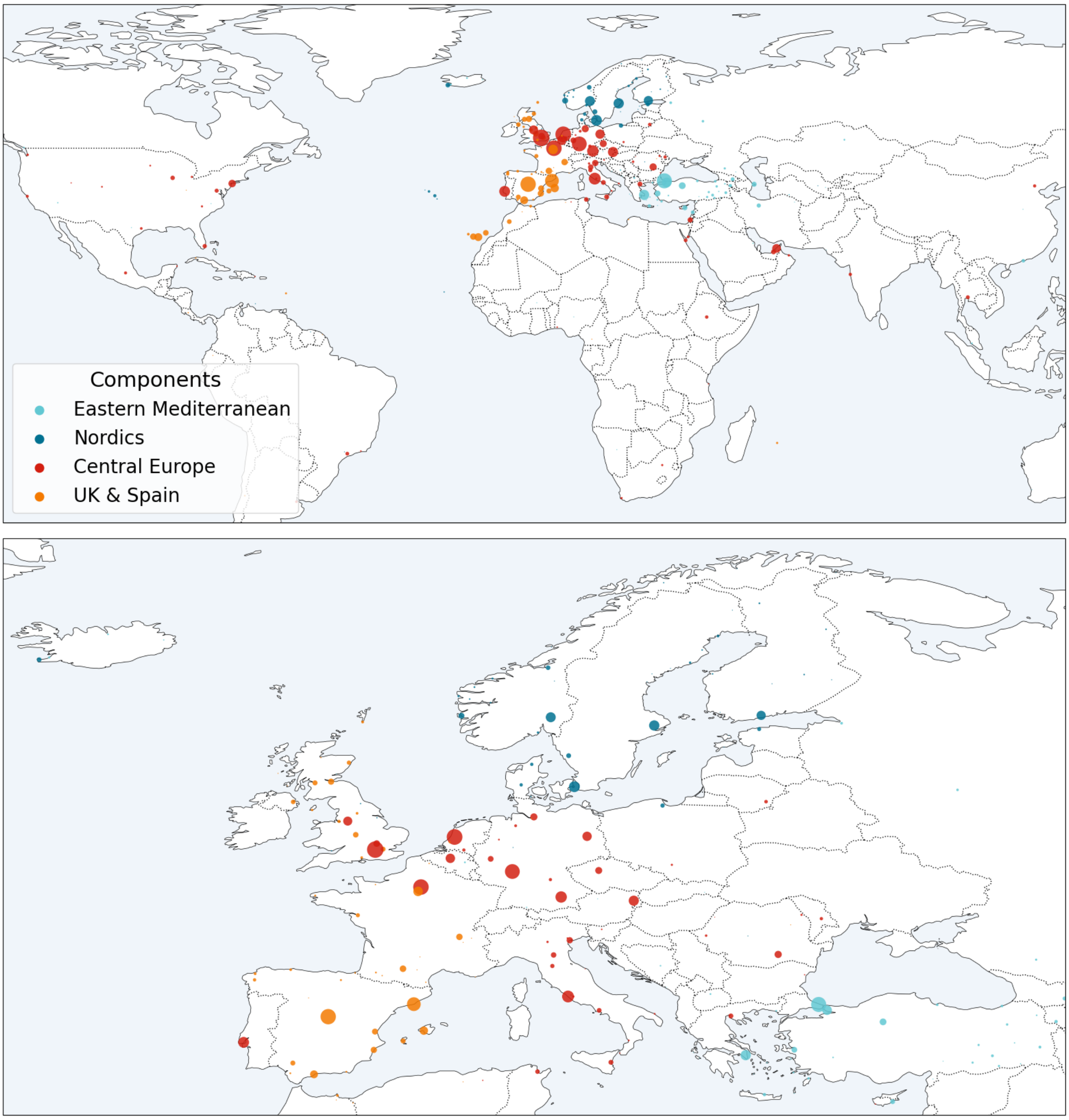

Figure 4.4: Visualization of the community assignments European Flights network for a sample month Map of the earth (Top) and a zoomed in version showing Europe only (Bottom) from dNOCD. Each node represents an airport, with node size proportional to its degree and color assigned as the maximum community affiliation for that node.

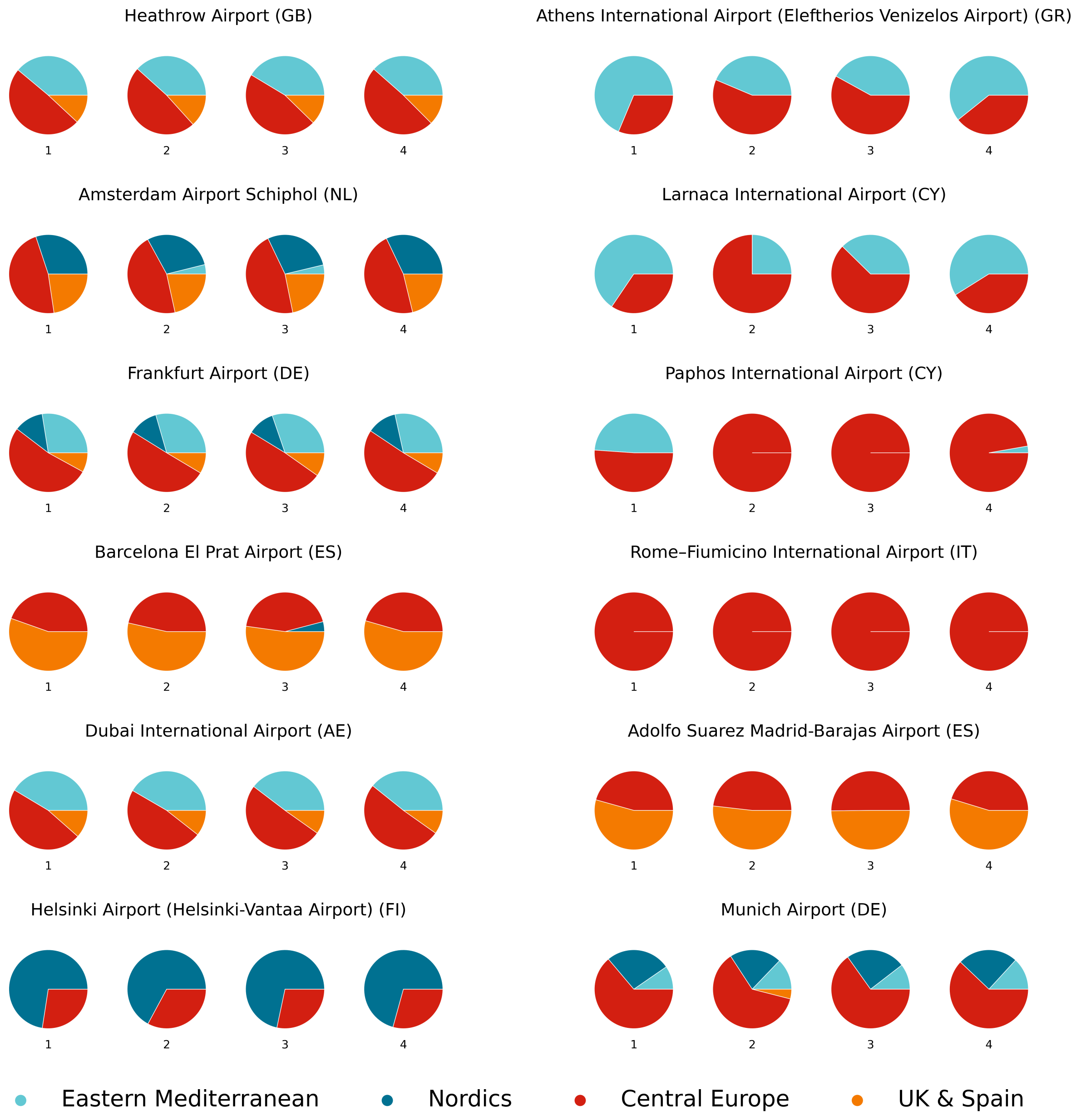

Figure 4.5: Pie charts with the % of community affiliations for some airports at each timestep using dNOCD.

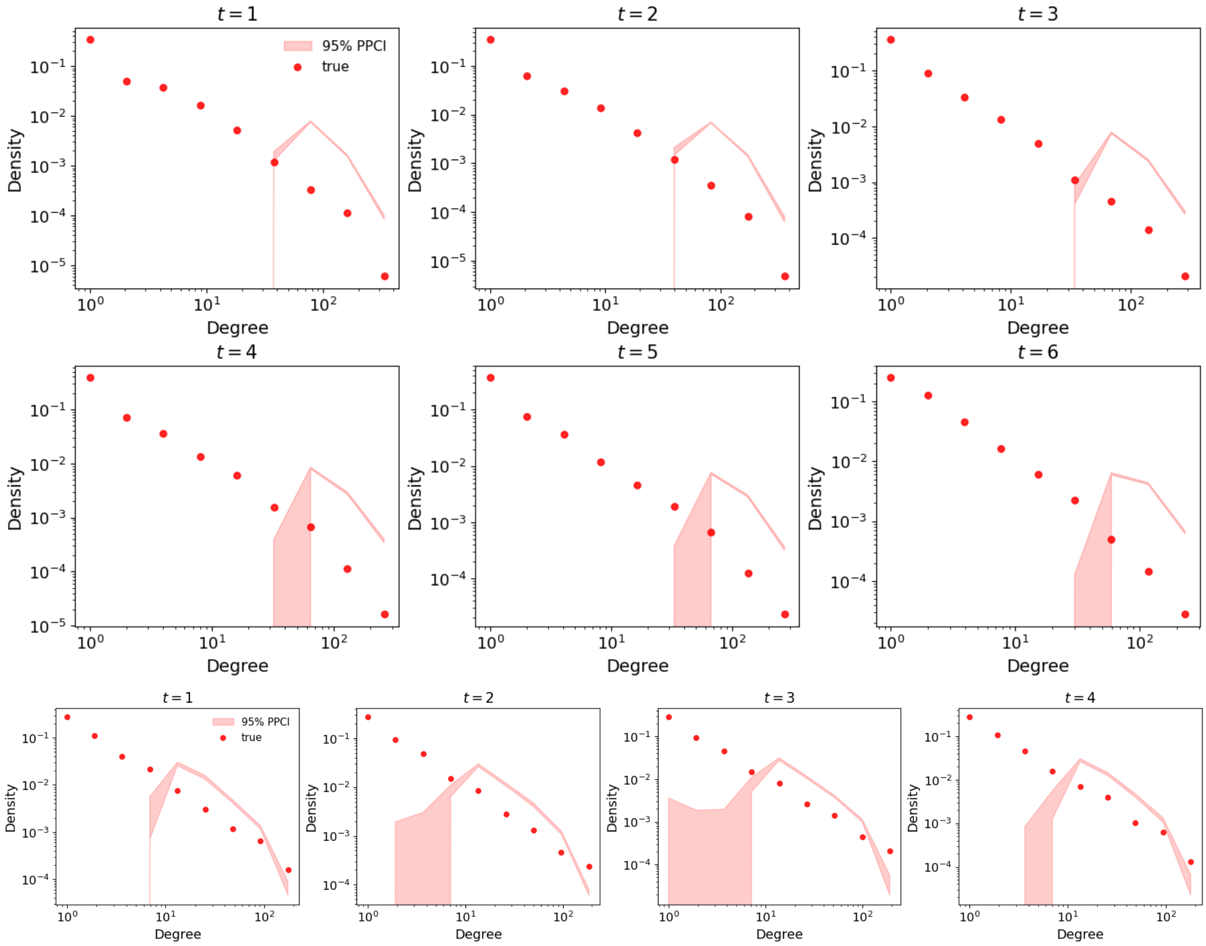

Figure 4.6: Degree distribution from dNOCD for the Reuters network (Top two rows) and the European Flights Network (Bottom row)

Additionally, the dNOCD architecture lacks an explicit mechanism for representing degree heterogeneity, in contrast to the dedicated base sociability weight w_{0i} that dynSNetOC draws from an etBFRY distribution and uses to absorb the power-law tail of the degree distribution. The most likely culprit in dNOCD is the batch normalization layer applied after the first graph convolution, following the original NOCD architecture [10]. Batch normalization rescales each feature dimension across nodes to have zero mean and unit variance, which by construction collapses the absolute scale of the activations and erases precisely the degree information necessary to scale $\mathbf{F}^{(t)}$.

5 Conclusion and Future Work

TABLE OF CONTENTS

5.1 Conclusion

This thesis set out to address the problem of dynamic overlapping community detection on sparse networks, with the goal of designing models that simultaneously capture the four properties that characterise the great majority of real-world networks: sparse connectivity, power-law degree distributions, overlapping community structure, and time-varying node memberships and edges. As discussed in Chapter 1, neither of the two dominant methodological paradigms for community detection currently provides a model that handles all four properties within a single framework. Bayesian nonparametric models built on the Caron-Fox construction capture sparsity and degree heterogeneity but, in their existing forms, do not jointly support overlap and temporal evolution. Graph neural networks scale to large data and can be trained end-to-end, but have not been extended to overlapping community detection in the dynamic setting. The contribution of this thesis is to close this gap from both directions, and, importantly, to evaluate the two resulting models on the same data so that the trade-offs between paradigms can be examined in a controlled way.

The first contribution, developed in Chapter 3, is *dynSNetOC*, a Bayesian nonparametric model for dynamic sparse graphs with overlapping communities. The model is constructed from vectors of completely random measures coupled through a latent Markov process governing the evolution of node affiliations, and observations are generated from a Bernoulli-Poisson likelihood. Inference is performed with the No-U-Turn Sampler, augmented by reparameterisation strategies that decouple the geometry of the sampled variables from the conditioning latent state and yield markedly improved sampling efficiency. The synthetic experiment confirms that the inference procedure recovers ground-truth sociability weights, affiliation scores, and hyperparameters of graphs drawn from the model itself, with 95% posterior predictive credible intervals covering the empirical degree distribution at every timestep. On the Reuters Terror corpus, dynSNetOC recovers four thematically coherent and temporally consistent communities that align with the historical narrative of the period following the September 11 attacks, with topics emerging and dissolving across weeks in a way that mirrors the actual unfolding of events. On the European Flights network, the recovered communities correspond to interpretable geographic clusters of airports. In contrast, both dynamic stochastic-blockmodel baselines, dynSBM and dMMSB, fail on both real datasets: dynSBM collapses onto a single community on the Flights data, and dMMSB reduces to a sociability-only diagonal solution. These failures confirm the central methodological hypothesis that dense blockmodel constructions cannot capture the joint sparsity, heavy-tailed degree behaviour, and overlapping structure of these networks.

The second contribution, developed in Chapter 4 , is *dNOCD*, a temporal extension of the NOCD graph neural network. Community affiliations are produced by a shared graph convolutional encoder applied at every timestep, with temporal coherence imposed through an L_2 smoothness penalty between consecutive affiliation matrices, and the network is trained end-to-end against the same Bernoulli-Poisson likelihood used by dynSNetOC. dNOCD shares the observation model and the high-level inferential target with dynSNetOC, but differs in three important respects: affiliations are produced by a deterministic neural encoder rather than sampled from a nonparametric prior; temporal dependence is enforced by an explicit regulariser rather than a latent Markov process; and inference is gradient-based, scaling on GPU hardware in time linear in the number of sampled edges and non-edges per step. On both real-world datasets dNOCD recovers communities that are qualitatively comparable to those of dynSNetOC, but it fails to reproduce their power-law degree distributions: the architecture has no explicit mechanism for representing degree heterogeneity as the batch-normalisation layer that follows the first graph convolution rescales node activations and erases the absolute-scale information needed to model the heavy tail and and the balanced Bernoulli–Poisson objective further discards degree signal. This is a genuine capability gap, not merely a tuning artefact, and it is the one property of the four motivating this thesis that dNOCD does not capture.

Evaluating dynSNetOC and dNOCD on the same two real-world datasets surfaces a clear methodological trade-off between the two paradigms, which is one of the central empirical contributions of this thesis. The Bayesian nonparametric model is more expensive at inference time, scaling with the cost of MCMC over the joint posterior, but in return delivers calibrated uncertainty in the form of posterior predictive intervals, interpretable hyperparameters governing sparsity and degree behaviour, and a fully generative description of the network that supports downstream tasks such as graph simulation and posterior predictive checks. The graph neural network model gives up uncertainty quantification and the explicit generative structure, but in return offers a substantial speed-up per training epoch, straightforward GPU parallelisation, and the ability to incorporate node features through the encoder. Which approach is preferable therefore depends on the downstream application: when interpretability, uncertainty, and a generative model of network growth are required, dynSNetOC is the natural choice; when scale, training time, or the use of side information are the priority (provided that faithful modelling of the degree distribution is not itself a requirement), dNOCD is more appropriate. The empirical similarity of the recovered communities on the Reuters and Flights datasets suggests that, on networks of moderate size, both paradigms can yield similar structure, and that the choice between them is driven by inferential and computational considerations rather than by the quality of the recovered communities themselves.

5.2 Future Work

The work presented in this thesis admits several natural extensions, both within each model and across the two paradigms. We outline the most promising directions below.

5.2.1 Incorporating Node Features into dynSNetOC

A clear methodological asymmetry between dynSNetOC and dNOCD is that the latter can readily condition on node features through the graph convolutional encoder, while dynSNetOC, in its current form, is purely structural. Extending dynSNetOC to incorporate node-level covariates would close this gap and bring the Bayesian model closer to the regime in which deep approaches typically dominate. A natural route is to let the base sociabilities w_{0i} or the affiliation scores $\beta_{ik}^{(t)}$ depend on covariates, while preserving the sparsity and power-law properties that the completely-random-measure construction provides.

5.2.2 Alternative Inference for dynSNetOC

The principal computational bottleneck of dynSNetOC is the No-U-Turn Sampler, whose cost scales unfavourably with the size of the network and the number of timesteps. Stochastic variational inference (SVI) is a natural candidate for a more scalable alternative. Rather than sampling from the posterior directly, SVI introduces a parametric variational family q_ϕ and optimises the evidence lower bound with respect to ϕ via gradient ascent on mini-batches of observations [45] (in this case, edge subsampling), converting a sampling problem into an optimisation problem that scales to large datasets. However, its application to dynSNetOC is non-trivial and warrants careful consideration. The core difficulty is that the latent variables of dynSNetOC live on the positive reals with heavy right tails that a standard mean-field Gaussian family approximates poorly. This matters because the power-law degree behaviour that distinguishes dynSNetOC from blockmodel baselines is encoded precisely in those tails: a variational family that underestimates them will systematically underestimate the sociability of the most structurally important nodes.

5.2.3 Degree-Corrected dNOCD

In Chapter 4 we observed that dNOCD recovers coherent communities on both Reuters and the Flights network but fails to reproduce their empirical power-law degree distributions. The cause is structural: the encoder $F^{(t)} = \mathrm{GCN}_\theta(A^{(t)}, X^{(t)})$ must absorb both community membership and degree heterogeneity into a single non-negative affiliation matrix, and the balanced Bernoulli-Poisson objective of Equation 4.4. further discards degree information by sampling edges and non-edges uniformly. A natural fix is to port the multiplicative factorisation of dynSNetOC into the neural setting by adding a separate non-negative per-node sociability head s_i, shared across timesteps, so that the rate becomes

$$\Lambda_{ij}^{(t)} = s_i\, s_j \sum_{k=1}^{p} \mathbf{F}_{ik}^{(t)}\, \mathbf{F}_{jk}^{(t)}$$

mirroring the $w_{0i}\, w_{0j} \sum_k \beta_{ik}^{(t)} \beta_{jk}^{(t)}$ decomposition of dynSNetOC. The sociability head s_i would be computed by explicitly incorporating degree information into the model.

5.2.4 Amortized Variational Inference for Overlapping Community Detection

A more ambitious direction is to replace the per-graph optimization of dNOCD with an amortized variational framework, enabling community structure to be inferred for an unseen graph in a single forward pass without retraining. The proposed architecture combines a graph convolutional encoder with the Bernoulli-Poisson generative model as its decoder. The encoder maps the adjacency matrix and any available node features to per-node, per-community mean and variance parameters, from which a non-negative affiliation matrix **F** is drawn. The decoder evaluates the Bernoulli-Poisson log-likelihood of the observed graph under **F**, forming an evidence lower bound (ELBO) that is optimised without ground-truth community labels. Trained once on a distribution of graphs, the model generalises inductively to new graphs, directly addressing the key limitation of dNOCD, which must be re-optimised from scratch for every new input graph.

BIBLIOGRAPHY

[1] M. E. J. Newman, “The structure and function of complex networks,” *SIAM Review*, vol. 45, no. 2, p. 167–256, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1137/S003614450342480

[2] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, p. 509–512, Oct. 1999. [Online]. Available: http://dx.doi.org/10.1126/science.286.5439.509

[3] P. Holme and J. Saramäki, “Temporal networks,” *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012, temporal Networks. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370157312000841

[4] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0378873383900217

[5] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, “Mixed membership stochastic blockmodels,” *Journal of Machine Learning Research*, vol. 9, no. 65, pp. 1981–2014, 2008. [Online]. Available: http://jmlr.org/papers/v9/airoldi08a.html

[6] E. P. Xing, W. Fu, and L. Song, “A state-space mixed membership blockmodel for dynamic network tomography,” *The Annals of Applied Statistics*, vol. 4, no. 2, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1214/09-AOAS311

[7] C. Matias and V. Miele, “Statistical clustering of temporal networks through a dynamic stochastic block model,” 2016. [Online]. Available: https://arxiv.org/abs/1506.07464

[8] F. Caron and E. B. Fox, “Sparse graphs using exchangeable random measures,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 79, no. 5, p. 1295–1366, Sep. 2017. [Online]. Available: http://dx.doi.org/10.1111/rssb.12233

[9] A. Todeschini, X. Miscouridou, and F. Caron, “Exchangeable random measures for sparse and modular graphs with overlapping communities,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 82, no. 2, p. 487–520, Mar. 2020. [Online]. Available: http://dx.doi.org/10.1111/rssb.12363

[10] O. Shchur and S. Günnemann, “Overlapping community detection with graph neural networks,” 2019. [Online]. Available: https://arxiv.org/abs/1909.12201

[11] M. Newman, *Networks*. Oxford University Press, 07 2018. [Online]. Available: https://doi.org/10.1093/oso/9780198805090.001.0001

[12] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velickovic, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," *CoRR*, vol. abs/2104.13478, 2021. [Online]. Available: https://arxiv.org/abs/2104.13478

[13] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo," 2011. [Online]. Available: https://arxiv.org/abs/1111.4246

[14] J. Kingman, "Completely random measures," *Pacific Journal of Mathematics*, vol. 21, no. 1, pp. 59–78, 1967.

[15] J. Lee, X. Miscouridou, and F. Caron, "A unified construction for series representations and finite approximations of completely random measures," *Bernoulli*, vol. 29, no. 3, Aug. 2023. [Online]. Available: http://dx.doi.org/10.3150/22-BEJ1536

[16] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," *CoRR*, vol. abs/1704.01212, 2017. [Online]. Available: http://arxiv.org/abs/1704.01212

[17] D. Jin, Z. Yu, P. Jiao, S. Pan, D. He, J. Wu, P. S. Yu, and W. Zhang, "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1149–1170, 2023.

[18] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, Q. Z. Sheng, and P. S. Yu, "A comprehensive survey on community detection with deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 4682–4702, 2024.

[19] E. Abbe, "Community detection and stochastic block models: Recent developments," *Journal of Machine Learning Research*, vol. 18, no. 177, pp. 1–86, 2018. [Online]. Available: http://jmlr.org/papers/v18/16-480.html

[20] C. Lee and D. J. Wilkinson, "A review of stochastic block models and extensions for graph clustering," *Applied Network Science*, vol. 4, no. 1, Dec. 2019. [Online]. Available: http://dx.doi.org/10.1007/s41109-019-0232-2

[21] T. Funke and T. Becker, "Stochastic block models: A comparison of variants and inference methods," *PLOS ONE*, vol. 14, no. 4, pp. 1–40, 04 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0215296

[22] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin, "A bayesian approach toward finding communities and their evolutions in dynamic social networks," 04 2009, pp. 990–1001.

[23] K. S. Xu and A. O. Hero, "Dynamic stochastic blockmodels for time-evolving social networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, p. 552–562, Aug. 2014. [Online]. Available: http://dx.doi.org/10.1109/JSTSP.2014.2310294

[24] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *Journal of the american Statistical association*, vol. 97, no. 460, pp. 1090–1098, 2002.

[25] P. D. Hoff, "Multiplicative latent factor models for description and prediction of social networks," *Computational and mathematical organization theory*, vol. 15, no. 4, pp. 261–272, 2009.

[26] M. Ward, J. AHLQUIST, and A. ROZENAS, "Gravity's rainbow: A dynamic latent space model for the world trade network," *Network Science*, vol. 1, 04 2013.

[27] D. Durante and D. Dunson, "Bayesian Logistic Gaussian Process Models for Dynamic Networks," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Kaski and J. Corander, Eds., vol. 33. Reykjavik, Iceland: PMLR, 22–25 Apr 2014, pp. 194–201. [Online]. Available: https://proceedings.mlr.press/v33/durante14.html

[28] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using bayesian non-negative matrix factorization," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 83, p. 066114, 06 2011.

[29] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2021. [Online]. Available: https://arxiv.org/abs/2003.05991

[30] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity based community detection with deep learning," 01 2016.

[31] Z. Wang, C. Wang, C. Gao, X. Li, and X. Li, "An evolutionary autoencoder for dynamic community detection," *Science China Information Sciences*, vol. 63, no. 11, p. 212205, 2020.

[32] P. Orbanz and D. M. Roy, "Bayesian models of graphs, arrays and other exchangeable random structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 437–461, 2015.

[33] T. Herlau, M. N. Schmidt, and M. Mø rup, "Completely random measures for modelling block-structured sparse networks," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/3937230de3c8041e4da6ac3246a888e8-Paper.pdf

[34] C. Naik, F. Caron, J. Rousseau, Y. W. Teh, and K. Palla, "Bayesian nonparametrics for sparse dynamic networks," in *Machine Learning and Knowledge Discovery in Databases*, M.-R. Amini, S. Canu, A. Fischer, T. Guns, P. Kralj Novak, and G. Tsoumakas, Eds. Cham: Springer Nature Switzerland, 2023.

[35] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/jax-ml/jax

[36] D. Phan, N. Pradhan, and M. Jankowiak, "Composable effects for flexible and accelerated probabilistic programming in numpyro," *arXiv preprint arXiv:1912.11554*, 2019.

[37] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *J. Mach. Learn. Res.*, vol. 20, pp. 28:1–28:6, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-403.html

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019. [Online]. Available: https://arxiv.org/abs/1912.01703

[39] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[40] M. Fey, J. Sunil, A. Nitta, R. Puri, M. Shah, B. Stojanovič, R. Bendias, A. Barghi, V. Kocijan, Z. Zhang, X. He, J. E. Lenssen, and J. Leskovec, "Pyg 2.0: Scalable learning on real world graphs," in *Temporal Graph Learning Workshop @ KDD 2025*, 2025. [Online]. Available: https://openreview.net/forum?id=DHHLkQvWqs

[41] C. Matias and V. Miele, *dynsbm: Dynamic Stochastic Block Models*, 2017. [Online]. Available: https://CRAN.R-project.org/package=dynsbm

[42] K. J. Dooley and S. R. Corman, "The dynamics of electronic media coverage," *Communication and terrorism: Public and media responses to https://www.public.asu.edu/~corman/dist/dooley_corman_comm_terror.pdf*, vol. 9, no. 11, pp. 121–135, 2002.

[43] J. Li, W. K. Cheung, J. Liu, and C. Li, "On discovering community trends in social networks," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1. IEEE, 2009, pp. 230–237.

[44] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

[45] M. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," 2013. [Online]. Available: https://arxiv.org/abs/1206.7051

APPENDICES

APPENDIX I

MCMC trace plots for synthetic data

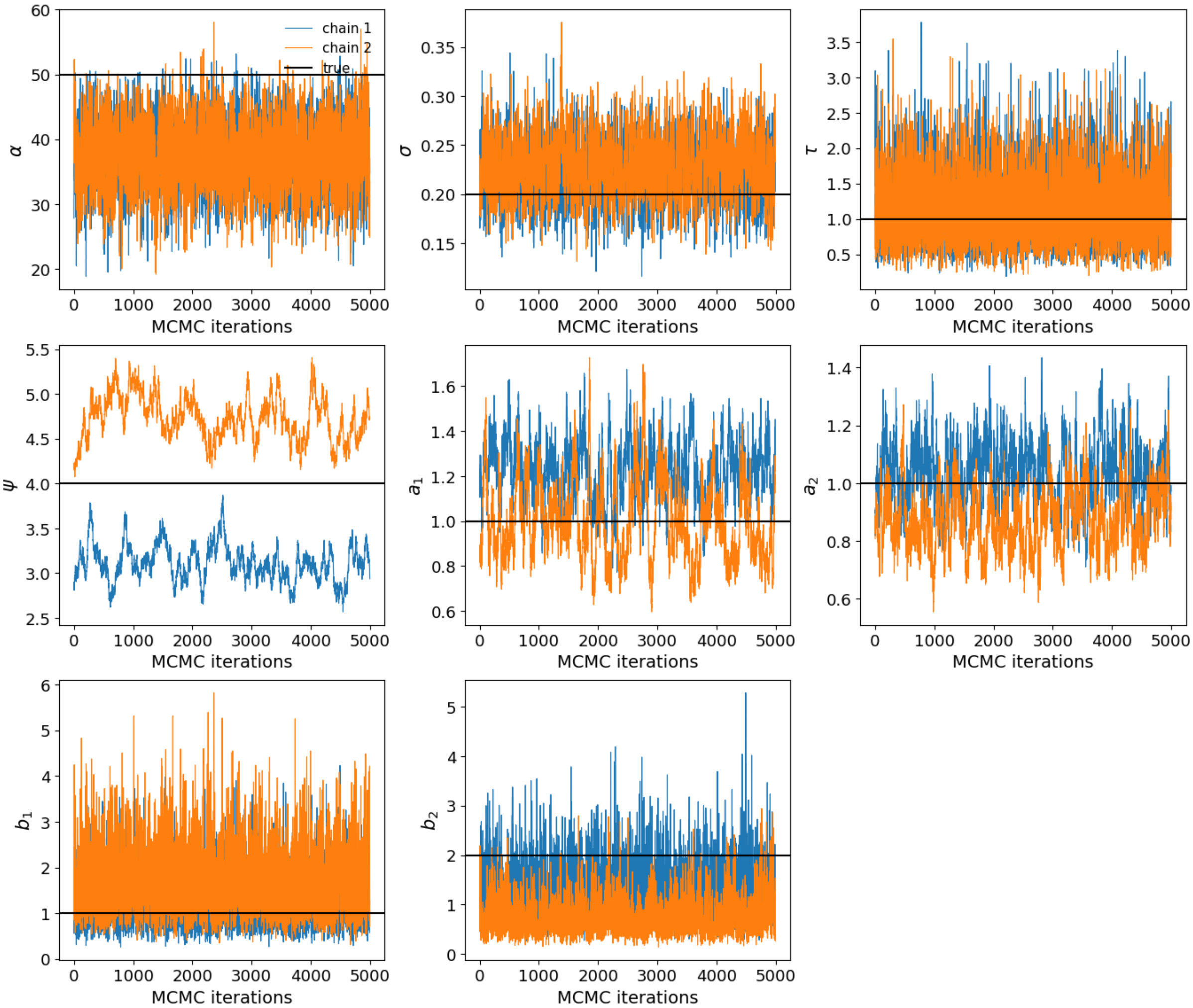

Figure I.1: MCMC trace plots for the parameters on the synthetic dataset

APPENDIX II

MCMC trace plots for Reuters Terror Dataset

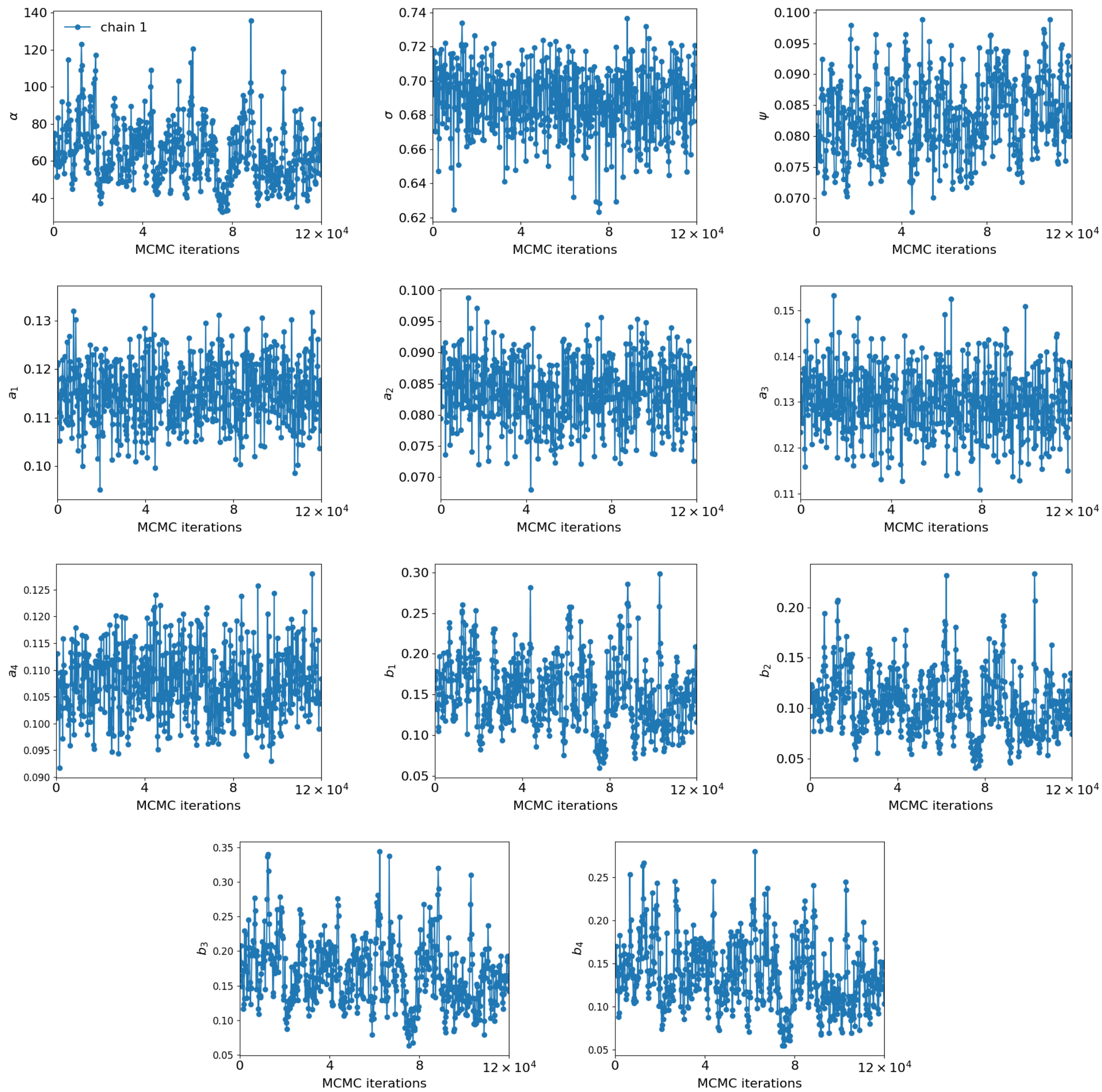

Figure II.1: MCMC trace plots of the parameters for the reuters dataset.

APPENDIX III

MCMC trace plots for the European Flights Network

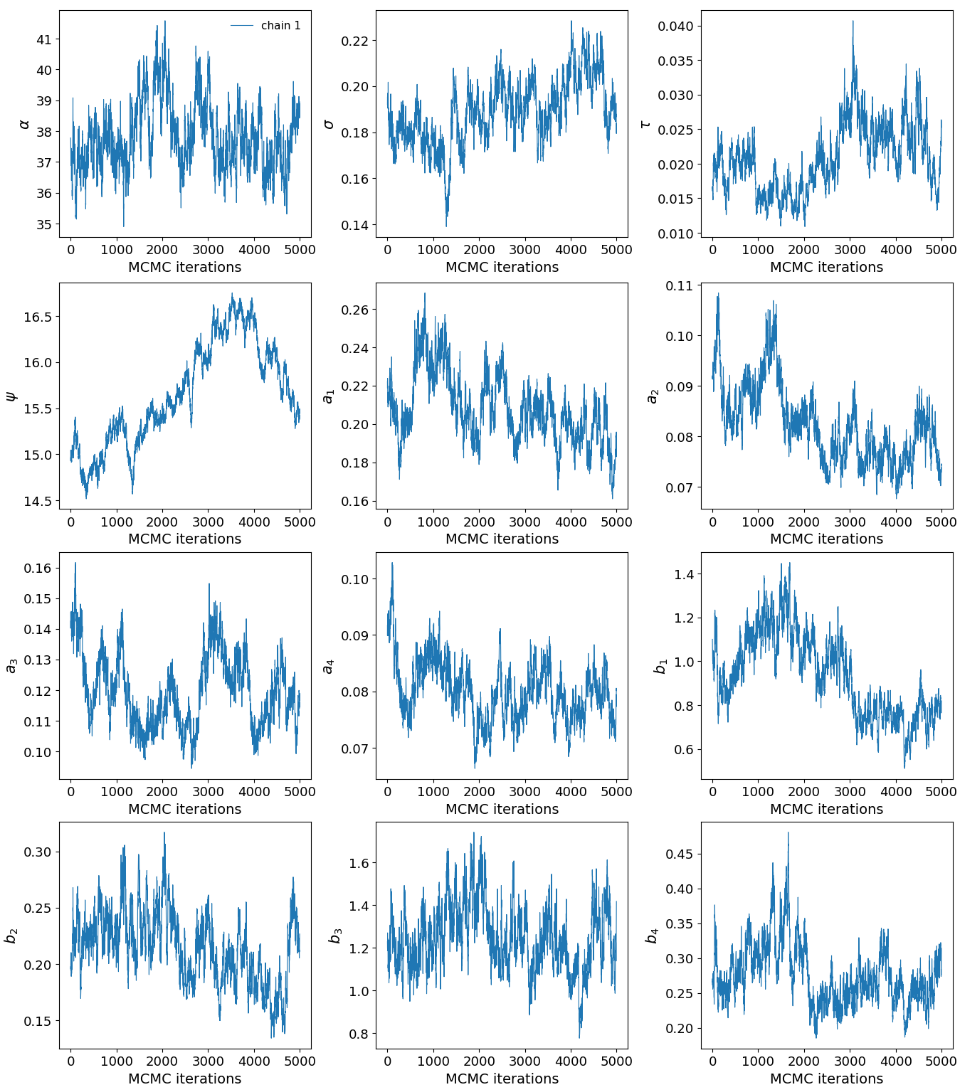

Figure III.1: MCMC trace plots for the parameters on the European Flights dataset