Decoupling Homophily and Reciprocity with Latent Space Network Models

Jiasen Yang
Department of Statistics
Purdue University
West Lafayette, IN
jiaseny@purdue.edu

Vinayak Rao
Department of Statistics
Purdue University
West Lafayette, IN
varao@purdue.edu

Jennifer Neville
Depts. of Computer Science and Statistics
Purdue University
West Lafayette, IN
neville@purdue.edu

Abstract

Networks form useful representations of data arising in various physical and social domains. In this work, we consider dynamic networks such as communication networks in which links connecting pairs of nodes appear over continuous time. We adopt a point process-based approach, and study latent space models which embed the nodes into Euclidean space. We propose models to capture two different aspects of dynamic network data: (*i*) communication occurs at a higher rate between individuals with similar features (homophily), and (*ii*) individuals tend to reciprocate communications from other nodes, but in a manner that varies across individuals. Our framework marries ideas from point process models, including Poisson and Hawkes processes, with ideas from latent space models of static networks. We evaluate our models over a range of tasks on real-world datasets and show that a *dual* latent space model, which accounts for heterogeneity in both reciprocity and homophily, significantly improves performance for both static and dynamic link prediction.

1 INTRODUCTION

Latent space models are a valuable tool for modeling social network data. By incorporating an embedding over nodes (*e.g.*, people), such models can account for unobserved preferences, interests, attitudes, etc. Typically, the likelihood of edges (interactions) between two nodes depends on their distance in the embedding space: the closer they are, the more likely they are to be linked. This reflects the notion of *homophily* (McPherson et al., 2001) that has been observed in many social domains: similar entities are more likely to form a tie than two randomly selected entities. As such, including latent spaces in models of static social networks has often improved descriptive and predictive accuracy with respect to modeling the link structure (*e.g.*, Hoff et al. 2002).

In this work, we focus on modeling the structure of dynamic networks, where interactions occur among entities over time. Dynamic networks are more complex than static networks because the temporal interactions can be varied and bursty, reflecting new, repeated, or correlated events. Much of the recent work in modeling temporal networks has typically represented the input networks as a sequence of snapshots taken at discrete time-points and often used Markov assumptions to restrict temporal correlations to the previous time-step.

It is much more natural to model the network dynamics using point processes, particularly when the interactions indicate events that occur in continuous time (*e.g.*, each pair of nodes has a sequence of interactions over time). Previous work have applied various point processes such as Poisson processes (*e.g.*, Iwata et al. 2013), renewal processes (*e.g.*, Min et al. 2011), and Hawkes processes (*e.g.*, Blundell et al. 2012) to modeling network data. Hawkes processes in particular have attracted a great deal of recent interest due to their capability to capture *reciprocity* in interaction data. Reciprocity refers to the act of responding to a particular action with the same type of action (Ekeh, 1974). For example, in social network interactions, if one person sends another a message, the likelihood that the other person will respond and send a message back in the near future increases. However, recent work has focused more on modeling reciprocity with specific individuals (or clusters of people) rather than modeling the dependencies among individuals that may influence reciprocity.

In this work, we bring the strength of latent space models for static networks to point process models for dynamic networks. We make the key observation that the latent dimensions of users which influence link formation may be different from the latent dimensions of users

which influence reciprocity. We refer to the former as the user's *homophily* latent space—dimensions which include preferences, interests, attitudes, etc. We refer to the latter as the user's *reciprocal* latent space—dimensions which include prosociality, agreeableness, level of self-monitoring, adherence to social norms, etc. Since the set of temporal interactions in dynamic networks consists of various types of events (some new, some instigated by other events), it is unlikely that all such interactions are governed by the same process. We conjecture that different latent space embeddings will help models to distinguish bursty events due to reciprocation from other types of interactions in a conversation.

To explore these issues, we propose a set of latent space point process models including a Poisson process-based model and multiple Hawkes process-based models with different latent space embeddings. We evaluate the utility of the various models both quantitatively and qualitatively through a set of carefully designed experiments on real-world datasets. Our results show that a *dual* latent space Hawkes process model, which contains latent spaces for both homophily and reciprocity, are more accurate for both dynamic and static link prediction. Moreover, the embeddings themselves can be used for subjective evaluation and provide insights on how various pairs of entities interact in the network.

Contributions. We make the following contributions:

- We propose and study a sequence of latent space point process models, including a Poisson process latent space model, two single-latent space Hawkes process models and a novel dual-latent space model.

- We develop methodology to evaluate the proposed models, including static and dynamic prediction tasks, and exploration of the learned embeddings.

- We evaluate the utility of our models both quantitatively and qualitatively on three real-world datasets. We show that incorporating both the homophily and reciprocal latent spaces improves predictive performance and gives rise to interpretable embeddings.

Outline. Section 2 reviews background on point processes and statistical network models; Section 3 formulates the problem and clarifies modeling assumptions; Section 4 proposes a set of latent space models for dynamic networks; Section 5 empirically evaluates the proposed models; Section 6 covers related work; and finally Section 7 concludes the paper.

2 BACKGROUND

2.1 POINT PROCESS MODELS

Point pattern data, consisting of the counts and locations of objects in some space, occur widely in the physical and social sciences. Our focus is on events occurring in time, so that the underlying space is the non-negative real line $\mathbb{R}_+$. We write realizations of such data as $\{N(t), t \geq 0\}$, where $N(t)$ is non-negative, integer, and non-decreasing, and gives the number of events occurring in the time interval $[0, t)$. The object $\{N(t), t \geq 0\}$ is sometimes called a counting measure, assigning to any interval $[s, t)$ a measure equal to the number of events in that interval. The special structure of the real line, as well as the causal nature of temporal dynamics have led to a variety of models for point process data on the real line. Two are particularly relevant to this paper:

Poisson Processes. The Poisson process is the canonical example of a point process. It is governed by a non-negative intensity (rate) function $\lambda(t)$, and has two properties: (*i*) for any $t > s$, the number of events within the time interval $[s, t)$, *i.e.*, $N(t) - N(s)$, follows a Poisson distribution with mean $\int_s^t \lambda(t)\,\mathrm{d}t$; and (*ii*) the number of events in disjoint time intervals are independent random variables. A Poisson process is homogeneous if its intensity function is constant ($\lambda(t) = \lambda$), otherwise it is referred to as non-homogeneous. If the intensity $\lambda(t)$ itself is random, then the point process is referred to as a Cox process (or a doubly-stochastic Poisson process).

Hawkes Processes. Hawkes processes have attracted much attention recently by capturing deviations from the Poisson process assumptions. Hawkes processes account for causality in the temporal dynamics of point pattern data by modeling self-excitation (when a single point process is involved), and mutual excitation or *reciprocity* (when collections of point processes are under study). The former is relevant to modeling individual activities over time (*e.g.*, hospital visits), while the latter is useful for modeling activities on communication networks (*e.g.*, email communications between members of an organization). In both examples, an initial event is often a trigger for a subsequent burst of activity.

Formally, a Hawkes process is a point process with conditional intensity function

$$\lambda(t|\mathcal{H}(t)) = \gamma + \int_0^t \phi(t - s)\,\mathrm{d}N(s)$$
$$= \gamma + \sum_{k:\, t_k < t} \phi(t - t_k) \qquad (1)$$

where $\mathcal{H}(t) = \{t_k : t_k < t\}$ consists of the event history at time t, γ is the base-rate, and $\phi(\cdot)$ is a *triggering* kernel that characterizes the excitatory effect that a past event has on the current event rate. For example, $\phi(\cdot)$ might be the exponential kernel $\phi(t) = \beta e^{-t/\tau}$, $t \geq 0$, implying that an event has an excitatory boost of magnitude β, which decays exponentially with a time scale τ. More generally, when we have m processes

$\{N_1(t), N_2(t), \cdots, N_m(t)\}$ that mutually excite one another, a multivariate Hawkes process has the conditional intensity of the j-th process given by

$$\lambda_j(t|\{\mathcal{H}_i(t)\}_{i=1}^m) = \gamma_j + \sum_{i=1}^m \int_0^t \phi_{ij}(t-s)\, \mathrm{d}N_i(s)\,.$$

Here $\mathcal{H}_i(t)$ denotes the event history associated with the i-th process $N_i(t)$ at time t; this consists of all events up to time t that are seen by process i.

We conclude with a technical remark. For any of these point processes, given a set of observed events $\{t_i\}_{i=1}^n$ in an interval $[0, T]$, the likelihood of a conditional intensity $\lambda(t)$ is given by

$$\mathcal{L}(\lambda(t)|\{t_i\}_{i=1}^n) = e^{-\Lambda(0,T)} \prod_{i=1}^n \lambda(t_i) \qquad (2)$$

where $\Lambda(0, T) = \int_0^T \lambda(t)\, \mathrm{d}t$ is the cumulative conditional intensity function. Different point process models make different independence assumptions about $\lambda(t)$.

2.2 STATISTICAL NETWORK MODELS

Graphs or networks, are useful representations of *relational* data natural to various physical, social, and informational domains. Mathematically, a graph G is written as $G = (V, E)$ where V is the set of vertices (nodes) and $E \subseteq V \times V$ is the set of edges (links). For example, in a network of individuals on a social media platform like Facebook, V represents users and E encodes undirected friendship relations. In a corporate email network, each node $v \in V$ might represent an employee in the corporation, and each edge (u, v), an email message sent from node u to node v. These two examples represent two different kinds of network data: static and dynamic.

Static Network Models. These models have a rich history, with proposed models including the Erdős-Rényi model (Erdős and Rényi, 1959), the preferential attachment model (Barabasi and Albert, 1999), the small-world model (Watts and Strogatz, 1998), the exponential random graph model (Holland and Leinhardt, 1981), and the stochastic blockmodel (Nowicki and Snijders, 2001), etc.

Of particular interest to us is the seminal work of Hoff et al. (2002), which models the probability p_{uv} of a link between two nodes u and v via a logistic regression model that depends on the observed features of nodes u and v, as well as the Euclidean distance $\|\mathbf{z}_u - \mathbf{z}_v\|_2$ between their *latent* features $\mathbf{z}_u, \mathbf{z}_v \in \mathbb{R}^d$. Young and Scheinerman (2007) proposed a similar model where the Euclidean distance $\|\mathbf{z}_u - \mathbf{z}_v\|_2$ is replaced by the dot-product $\mathbf{z}_u^\mathsf{T} \mathbf{z}_v$. An advantage of latent space models is that by estimating the latent features $\mathbf{z}_v$ for each node v, we obtain a mapping that *embeds* the nodes of the graph

into a Euclidean space $\mathbb{R}^d$. Such an embedding is much more amenable to conventional statistical analysis.

Dynamic Network Models. In many real-world applications, the network structure evolves over time, and we have access to fine-grained *temporal* information describing the evolution of the network structure. For instance, new users join a social network like Facebook every day; additionally, existing users may be connected by newly forged friendships. Similarly, in a corporate email network, the server records the precise time-stamps of every message sent between every pair of nodes.

In contrast to the extensive literature on static networks, statistical models for dynamic networks are much less explored. Existing models (Sarkar and Moore, 2005; Miller et al., 2009; Fan and Shelton, 2009; Fu and Xing, 2009; Hanneke and Xing, 2010; Snijders et al., 2010; Durante and Dunson, 2014) typically assume that the available data contain a sequence of graph snapshots captured at discrete time-points, and that the network evolution follows Markov transition. Such approximations discard important information when one has exact time-stamps for each link event, and require modelers to choose a particular temporal resolution to study network dynamics. A much more natural approach is to merge ideas from point process modeling with network models.

3 PROBLEM DEFINITION

In this work, we develop and explore latent space models of dynamic network data. We consider network data with the following properties:

- There exists a fixed set of vertices $V = \{1, \ldots, n\}$ throughout an observation time period $[0, T]$.

- For each ordered pair of vertices (u, v), we observe a set of event-times, corresponding to a sequence of directed links or messages from u to v. We write the overall observed data as $\{(u, v, \mathcal{H}_{uv})\}_{u,v \in V}$, where $\mathcal{H}_{uv} \triangleq \{t_i^{uv}\}_{i=1}^{n_{uv}}$ records the set of all time-points at which u sent v a message. We write $n_{uv} \geq 0$ for the total number of messages from u to v.

- A node never sends a message to itself; and the granularity of measurements is fine enough that the probability of two simultaneous events is zero.

These properties naturally motivate a point process-based approach, and we model the arrival times $\{t_i^{uv}\}$ of each link from node u to v as realizations of a point process $N_{uv}(t)$, $t \in [0, T]$. The dynamic network evolving over time consists of $n^2 - n$ point processes $N_{uv}(t)$, which if treated as independent, involves $\mathcal{O}(n^2)$ parameters. Such an independence assumption however ignores important structure in the dynamics of the point processes. We assume two sources of dependency:

Static dependencies due to homophily, where baseline event rates vary between pairs of nodes because of shared features. Among other things, this accounts for the fact that the two processes $N_{uv}(t)$ and $N_{uw}(t)$ have a node u in common and therefore will share statistical properties. In general, homophily reflects how similarity in node-level properties (such as preferences, interests, and attitudes) affects link formation.

Dynamic interactions due to reciprocity, where activity between pairs of nodes is a function of previous history. At its simplest, this might account for reciprocity in communications between a pair of individuals. More generally, this accounts for how social influence, charisma, and the user-role affects the dynamics in a sequence of interactions. The nature of this reciprocation might depend on shared features between two nodes different from the features relevant to homophily.

Inspired by the work of Hoff et al. (2002), we model these phenomena by assigning to each node $v \in V$ a set of latent features. For the first effect, we write its feature vector as $\mathbf{z}_v \in \mathbb{R}^d$, and assume that the intensity function $\lambda_{uv}(t)$ underlying the process $N_{uv}(t)$ depends on the Euclidean distance between $\|\mathbf{z}_u - \mathbf{z}_v\|_2$. To account for the second effect, we cannot assume that $\lambda_{uv}(t)$ is fixed in time given these latent features, and instead must allow it to depend on previous network activity. This dependency will again be described by latent features associated with each node, but a different set which we write as $\mathbf{x}_v \in \mathbb{R}^d$.

While we do not explicitly assume the availability of any observed features for each node, they can be directly incorporated in our models by augmenting the $\mathbf{x}$- and $\mathbf{z}$-vectors. We also do not assume any additional information (such as message text or topic) for each link apart from its time-stamp, but we note that such information can be utilized by augmenting the hierarchical generative models with another level, as demonstrated in *e.g.*, He et al. (2015); Tan et al. (2016).

4 LATENT SPACE POINT PROCESS MODELS OF DYNAMIC NETWORKS

In this section we present a series of latent space point process models for dynamic network data. We begin with the most straightforward model that only captures homophily, and proceed through a sequence of models of increasing complexity.

4.1 POISSON LATENT SPACE MODEL

Perhaps the simplest latent space network point process model treats messages from a node u to v as a time-homogeneous Poisson process whose intensity is a function of the Euclidean distance between them in a latent feature space. In equations:

Poisson-rate Latent Space (PLS) Model

$$\mathbf{z}_v \sim \mathcal{N}(\mathbf{0}, \sigma^2\, \mathbf{I}_{d \times d}) \qquad \forall v \in V$$

$$\lambda_{uv}(t) = \gamma\, e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2} \qquad \forall u \neq v$$

$$N_{uv}(\cdot) \sim \mathsf{PoissonProcess}(\lambda_{uv}(\cdot)) \qquad \forall u \neq v$$

Here, we have placed independent Gaussian priors on the latent features for each node, resulting in a collection of correlated doubly-stochastic Poisson processes. The parameter γ can be assigned a prior if we have node-level or edge-level covariates available, but for identifiability we tie the parameter across all pairs of nodes.

4.2 HAWKES LATENT SPACE MODELS

The remaining models augment the latent-space representation with additional non-Poissonian dynamics that capture reciprocity in communications across a network.

Hawkes Process Model. At its simplest, a node v is much more likely to send node u a message if u had just sent v a message earlier. To incorporate such reciprocity, the intensity function $\lambda_{uv}(t)$ governing the $N_{uv}(t)$ process can be modeled to depend on the events history of the reciprocal process $N_{vu}(t)$. Hawkes processes provide a simple mathematical tool to achieve this.

Specifically, for nodes $u, v \in V$, $u \neq v$, we model the pair of processes $N_{uv}(t)$ and $N_{vu}(t)$, as a bivariate Hawkes process, with intensity depending on the event histories, $\mathcal{H}_{uv} \triangleq \{t_i^{uv}\}_{i=1}^{n_{uv}}$ and $\mathcal{H}_{vu} \triangleq \{t_i^{vu}\}_{i=1}^{n_{vu}}$:

$$\lambda_{uv}(t|\mathcal{H}_{uv}, \mathcal{H}_{vu}) = \gamma_{uv} + \sum_{k:\, t_k^{vu} < t} \phi_{uv}(t - t_k^{vu}). \quad (3)$$

We have removed the self-excitation component since we do not consider self-loops in the network. Similar approaches have appeared in previous work (*e.g.*, Blundell et al. 2012), but we will comment on these in Section 6. While it is standard to parametrize the triggering function $\phi_{uv}(\cdot)$ as an exponential kernel with time scale τ, we found that learning τ suffered from identifiability issues. Instead, we model $\phi_{uv}(\cdot)$ as a weighted combination of basis kernels:

$$\phi_{uv}(t) = \sum_{b=1}^{B} \xi_b^{uv} \phi_b(t) \quad (4)$$

where ξ_b^{uv} is the weight of the kernel ϕ_b. We consider two possible forms for the basis kernel ϕ_b: (*i*) exponential kernels with length-scale τ, $\phi_b(t) = e^{-t/\tau}$; and (*ii*) locally periodic kernels with period p and length-scale τ, $\phi_b(t) = e^{-t/\tau} \sin^2\left(\frac{\pi t}{\tau}\right)$. In our experiments, we utilize kernels with time-scales of an hour, a day, and a week, which are interpretable and realistic for our applications.

We summarize the Hawkes process model below:

Hawkes Process (HP) Model

$$\lambda_{uv}(t) = \gamma + \sum_{k:\, t_k^{vu} < t} \sum_{b=1}^{B} \xi_b\, \phi_b(t - t_k^{vu}) \quad \forall u \neq v$$

$$N_{uv}(\cdot) \sim \mathsf{HawkesProcess}(\lambda_{uv}(\cdot)) \qquad \forall u \neq v$$

We have again tied the parameters $\gamma_{uv} \equiv \gamma$ and $\xi_b^{uv} \equiv \xi_b$ across all node-pairs to avoid identifiability issues.

Hawkes Base-Rate Latent Space Model. The most straightforward way of modeling both homophily and reciprocity is to add the Hawkes triggering function term to the intensity functions of the previous PLS model:

Hawkes Base-Rate Latent Space (BLS) Model

$$\mathbf{z}_v \sim \mathcal{N}(\mathbf{0}, \sigma^2\, \mathbf{I}_{d \times d}) \qquad \forall v \in V$$

$$\lambda_{uv}(t) = \gamma\, e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$$

$$+ \sum_{k:\, t_k^{vu} < t} \sum_{b=1}^{B} \xi_b\, \phi_b(t - t_k^{vu}) \quad \forall u \neq v$$

$$N_{uv}(\cdot) \sim \mathsf{HawkesProcess}(\lambda_{uv}(\cdot)) \qquad \forall u \neq v$$

Here, and with the Poisson-rate latent space model, we shall refer to the $\mathbf{z}$-space as the *homophily latent space*, with the distance between $\mathbf{z}_u$ and $\mathbf{z}_v$ reflecting how dissimilar u and v are, regardless of their communication history. This distance sets a baseline rate of communication between the two nodes—*i.e.*, the rate at which one node initiates communication with the other. The Hawkes component captures the fact that having initiated a new communication, subsequent messages in that thread will follow different dynamics.

Hawkes Reciprocal Latent Space Model. The previous model assumes heterogeneity only in the rate at which different node-pairs initiate communications, and the Hawkes dynamics are themselves assumed to be homogeneous across all pairs. Our next model modifies Eq. (3) to reverse this assumption, associating latent features with reciprocity rather than the base-rate:

Hawkes Reciprocal Latent Space (RLS) Model

$$\mathbf{x}_v \sim \mathcal{N}(\mathbf{0}, \sigma^2\, \mathbf{I}_{d \times d}) \qquad \forall v \in V$$

$$\lambda_{uv}(t) = \gamma + \sum_{k:\, t_k^{vu} < t} \sum_{b=1}^{B} \xi_b\, e^{-\|\mathbf{x}_u - \mathbf{x}_v\|_2^2}\, \phi_b(t - t_k^{vu})$$

$$N_{uv}(\cdot) \sim \mathsf{HawkesProcess}(\lambda_{uv}(\cdot)) \qquad \forall u \neq v$$

We shall refer to the $\mathbf{x}$-space as the *reciprocal latent space*, since it modulates the magnitude of excitation triggered by each message between the pair of nodes.

Hawkes Dual Latent Space Model. As a final model, we combine the ideas of homophily and reciprocal latent spaces into a single model. Our final Hawkes process latent space model accounts for heterogeneity both in how two users initiate communications, as well as in the dynamics within a particular exchange. Using a mixture of exponential and periodic kernels with various length-scales, we can investigate whether a message sent from node u to v is more likely to trigger an immediate response, a response sometime over a week, or whether communications have a periodic nature.

Hawkes Dual Latent Space (DLS) Model

$$\mathbf{z}_v \sim \mathcal{N}(\mathbf{0}, \sigma^2\, \mathbf{I}_{d \times d}) \qquad \forall v \in V$$

$$\boldsymbol{\mu}_v \sim \mathcal{N}(\mathbf{0}, \sigma_\mu^2\, \mathbf{I}_{d \times d}) \qquad \forall v \in V$$

$$\boldsymbol{\varepsilon}_v^{(b)} \sim \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2\, \mathbf{I}_{d \times d}) \qquad \forall v \in V,\ b = 1, \ldots, B$$

$$\mathbf{x}_v^{(b)} \sim \boldsymbol{\mu}_v + \boldsymbol{\varepsilon}_v^{(b)} \qquad \forall v \in V,\ b = 1, \ldots, B$$

$$\lambda_{uv}(t) = \gamma\, e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$$

$$+ \sum_{k:\, t_k^{vu} < t} \sum_{b=1}^{B} \beta\, e^{-\|\mathbf{x}_u^{(b)} - \mathbf{x}_v^{(b)}\|_2^2}\, \phi_b(t - t_k^{vu})$$

$$N_{uv}(\cdot) \sim \mathsf{HawkesProcess}(\lambda_{uv}(\cdot)) \qquad \forall u \neq v$$

Notice that we have also placed a hierarchical prior on the B reciprocal latent spaces to enforce consistency between the learned latent features across different kernels.

4.3 INFERENCE

For all the models we have discussed, we perform maximum a posteriori (MAP) inference over the unknown parameters. Recall that we place independent standard Gaussian priors on the latent space vectors $\{\mathbf{z}_v\}_{v \in V}$ and $\{\mathbf{x}_v\}_{v \in V}$. Additionally, we place Gamma priors on the base rate γ and triggering magnitudes $\{\xi_b\}_{b=1}^{B}$ and β. Inference is tractable since it follows from Eq. (2) that the log-likelihood of all communications observed over the entire network $\{(u, v, \{t_i^{uv}\}_{i=1}^{n_{uv}})\}_{u,v \in V}$ can be written as

$$\log \mathcal{L} = \sum_{\substack{u,v=1 \\ u \neq v}}^{n} \left\{ -\Lambda_{uv}(0, T) + \sum_{i=1}^{n_{uv}} \log \lambda_{uv}(t_i^{uv}) \right\} \quad (5)$$

where the intensities $\lambda_{uv}(t)$ are specified for each model, and the cumulative intensities can be found in closed-form by noticing that for the basis kernel ϕ_b, we have

$$\int_0^T \sum_{k:\, t_k^{vu} < t} \phi_b(t - t_k^{vu})\, \mathrm{d}t = \sum_{k=1}^{n_{vu}} [\Phi_b(T - t_k^{vu}) - \Phi_b(0)]$$

where $\Phi_b(t) \triangleq \int_0^t \phi_b(s)\,\mathrm{d}s$. In fact, the right-hand side, as well as the quantities $\{\sum_{k:\,t_k^{vu}<t_i^{uv}} \phi_b(t_i^{uv}-t_k^{vu})\}_{i=1}^{n_{uv}}$ are data statistics that can be pre-computed and cached for each pair of nodes $u, v \in V$ and kernel ϕ_b. Furthermore, the gradients of the log-posterior function are also available in closed form, and the optimization can be carried out using L-BFGS-B (Byrd et al., 1995). Detailed calculations are provided in the supplementary material.

We conclude this section with a brief summary of the complexity of each proposed model. Assuming that the number of nodes n and the dimensionality of the latent spaces d are both much larger than the number of basis-kernels B, the HP model has $\mathcal{O}(B)$ parameters, while the PLS, BLS, and RLS models have $\mathcal{O}(n \cdot d)$ parameters, and the DLS model has $\mathcal{O}(n \cdot d \cdot B)$ parameters.

5 EXPERIMENTAL EVALUATION

In this section, we evaluate the proposed models of Section 4, both quantitatively and qualitatively, on three real-world datasets. For the quantitative component, we evaluate model performance across multiple tasks, including predictive log-likelihood, dynamic link prediction, and (static) link prediction using the learned embeddings. For the qualitative component, we visualize the learned network embeddings, and demonstrate how the reciprocal latent spaces in the DLS model can be used to characterize different reciprocation patterns.

Dataset Description. We perform experiments on three real-world communication networks:

ENRON This is the "core" network of the Enron email dataset (Klimmt and Yang, 2004) consisting of communications among 155 Enron executives. Each node represents an employee, while each link corresponds to an email message. We consider the period between January 2000 and April 2002, during which the vast majority of communications occurred. The resulting dataset contains 9,646 email messages spanning a period of 453 days.

EMAIL This dataset contains email communications within Purdue University from July 2011 to February 2012. Each node in the network represents an email address, while each link corresponds to an email message. We cleaned up this dataset by filtering out mailing-lists, and extracted the one hundred nodes with largest total degree. The resulting network consists of 34,438 email messages spanning a period of 237 days.

FACEBOOK This dataset contains Facebook wall messages among students of Purdue University from March 2007 to March 2008. Each node in the network represents an anonymized user account, and

each link corresponds to a wall message. To focus on the "core" part of the network, we take a subset of the one hundred accounts with largest total degree. The resulting network consists of 18,865 wall messages posted over a period of 385 days.

Experiment Setup. For each network dataset, we sort the messages according to their timestamps, and split the dataset into a training set consisting of the first 70% messages, and a test set consisting of the remaining 30% messages. All models are trained on the training set, and all evaluation tasks are performed on the test set.

For all Hawkes process-based models (*cf.* Section 4.2), we utilize $B = 4$ basis kernels—three exponential kernels with length-scales one hour, one day, and one week, respectively: $\phi_1(t) = e^{-\frac{t}{1/24}}$, $\phi_2(t) = e^{-t}$, $\phi_3(t) = e^{-t/7}$; and a locally periodic kernel with both period and length-scale set to one week: $\phi_4(t) = e^{-t/7}\sin^2\left(\frac{\pi t}{7}\right)$ (all units are in days). For all latent space models, we set the dimensionality of the latent vectors to be $d = 100$.[1] For the BLS, RLS, and DLS models, we set $\sigma^2 = \sigma_\mu^2 = \sigma_\epsilon^2 = 1$. For MAP inference, we perform optimization using the L-BFGS-B solver in the SciPy package with analytical gradients derived for each model.

5.1 PREDICTIVE LOG-LIKELIHOOD

We learn the parameters of all models on the training set, and compute their predictive log-likelihood values on the test-set. From Table 1, we observe that the Hawkes process-based models significantly outperform the Poisson-rate latent space model (PLS), while the test log-likelihood values improve as we move from the base-rate latent space (BLS) and reciprocal latent space (RLS) models to the dual latent space (DLS) model. The DLS model also comfortably outperforms the Hawkes process (HP) model on two of the three datasets. For the Enron dataset, the simple HP model slightly outperforms the other models: we believe this is because the dataset is relatively unstructured, with pairwise reciprocity dominating most exchanges. Also notice that BLS outperforms PLS, which indicates that going beyond homogeneous Poisson processes to model reciprocity in the network indeed yields better predictive performance.

5.2 DYNAMIC LINK PREDICTION

We further gauge the performance of the learned models in a temporal link prediction task. Specifically, we randomly sample 100 time-points t_i during the test period, and ask every model to predict the probability that a link

---

[1] Since the DLS model contains one homophily latent space and four reciprocal latent spaces, one could argue that the latent spaces in the other models should be $(5\,d)$-dimensional. We experimented with setting $d = 500$ dimensions for all the other models, and obtained similar results to the $d = 100$ setting.

Table 1: Predictive log-likelihood.

Model	ENRON	EMAIL	FACEBOOK
HP	**-16226.155**	-2129.940	-7871.895
PLS	-37803.978	-112684.130	-66742.379
BLS	-21779.686	-9850.932	-12119.869
RLS	-16565.449	-2113.254	-7867.870
DLS	-16422.946	**185.264**	**-6421.609**

will appear between each pair of nodes in the $[t_i, t_i + \delta)$ time window (we set δ to be two weeks). Note that all models are equipped with parameters estimated from the training set, and for Hawkes process models we also condition on all the historical training and test events up to time t_i. For each time-point t_i, we then compute the area under the ROC curve (AUC) measured across all pairs of nodes according to the predicted probabilities given by each model.[2] Finally, we report the mean and standard deviations of the AUC scores across all 100 randomly sampled testing time-points in Table 2.

Table 2: Dynamic link prediction AUC scores.

Model	ENRON	EMAIL	FACEBOOK
HP	0.750 (0.070)	0.881 (0.088)	0.931 (0.095)
PLS	0.681 (0.041)	0.843 (0.087)	0.874 (0.078)
BLS	0.738 (0.065)	0.868 (0.095)	0.927 (0.096)
RLS	0.750 (0.070)	0.881 (0.088)	0.931 (0.095)
DLS	**0.928 (0.018)**	**0.971 (0.006)**	**0.979 (0.008)**

5.3 NETWORK EMBEDDING

As mentioned in Section 2.2, a major motivation for developing latent-space network models is that the learned latent feature vectors for each node effectively provide a mapping that embeds the observed network into Euclidean space. To evaluate the quality of the learned embeddings for each latent space model, we perform link prediction on the test set by collapsing the messages into a single undirected and unweighted graph, where there exists an edge between two nodes if at least one communication exists between them in the test set. Given the learned latent feature vectors $\{\mathbf{z}_v\}_{v \in V}$ (or $\{\mathbf{x}_v^{(b)}\}_{v \in V}$ for reciprocal latent spaces), we compute the predicted probability that an edge exists in the test graph via $p_{uv} \propto e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$, $\forall u, v \in V$, and then measure the link prediction AUC scores for all pairs of nodes.

In addition to the latent space models proposed in this work, we also compare with two popular approaches for embedding static networks:

Spectral Laplacian eigenmaps are widely used in spectral clustering (see *e.g.*, von Luxburg, 2007). Given the adjacency matrix $\mathbf{A}$ of the training network, we

compute the d eigenvectors corresponding to the smallest eigenvalues of the symmetric normalized Laplacian $\mathbf{L}^{\text{sym}} \triangleq \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is a diagonal matrix of node degrees.

node2vec This is a state-of-the-art deep learning approach to learning continuous feature representations for networks (Grover and Leskovec, 2016).[3]

For both Laplacian eigenmaps and node2vec, we form an adjacency matrix of the training network $\mathbf{A}$ by collapsing the messages in the training set into an undirected graph with each edge weighted by the number of communications between the corresponding pair of nodes.[4] We set $d = 100$ for fair comparison.

Table 3 shows the obtained AUC scores, and Figure 1 plots the corresponding ROC curves. Shown alongside the ROC curves are two-dimensional projections (obtained using PCA) of the 100-dimensional latent spaces learned using each method. We observe that DLS performs on par with node2vec, and outperforms all other approaches in terms of AUC score.[5]

Table 3: Static link prediction AUC scores.

Model	ENRON	EMAIL	FACEBOOK
PLS	0.512	0.483	0.505
BLS	0.512	0.483	0.505
RLS	0.601	0.295	0.445
DLS	**0.906**	**0.958**	0.947
Spectral	0.687	0.428	0.452
node2vec	0.829	**0.958**	**0.956**

Homophily and Reciprocal Latent Spaces. For the DLS model, we found that the learned homophily latent spaces $\{\mathbf{z}_v\}_{v \in V}$ always perform much better than the reciprocal latent spaces $\{\{\mathbf{x}_v^{(b)}\}_{v \in V}\}_{b=1}^B$ under the static link prediction setup, as shown in the ROC curves for DLS-$\mathbf{z}$ and DLS-$\mathbf{x}^{(1)}$ in Figure 1.[6] Moreover, simply augmenting the homophily latent space with the reciprocal latent spaces actually leads to degraded performance in link prediction AUC. However, notice that the BLS model actually corresponds to a DLS model where

---

[2]The predicted probability of node u sending v at least one message during the time interval $[t, t + \delta)$ can be computed as $1 - \exp\{-\int_t^{t+\delta} \lambda_{uv}(s)\,\mathrm{d}s\}$.

[3]We utilize the publicly available implementation at `http://snap.stanford.edu/node2vec/`.

[4]For both Laplacian eigenmaps and node2vec, we have also experimented with treating the adjacency matrix $\mathbf{A}$ as binary (unweighted), but both methods exhibit degraded performance.

[4]For the DLS model, the homophily latent space is used.

[5]Notice that the current experiment setup does not yield standard errors for the AUC scores, since there is only a single training/test set split. To investigate the statistical significance of the results, we conducted a further experiment which showed that while DLS significantly outperforms node2vec on ENRON, their performance are comparable on EMAIL and FACEBOOK. See the supplementary material for details.

[6]The other reciprocal latent spaces exhibit similar performance, and we omit them from the plots to reduce clutter.

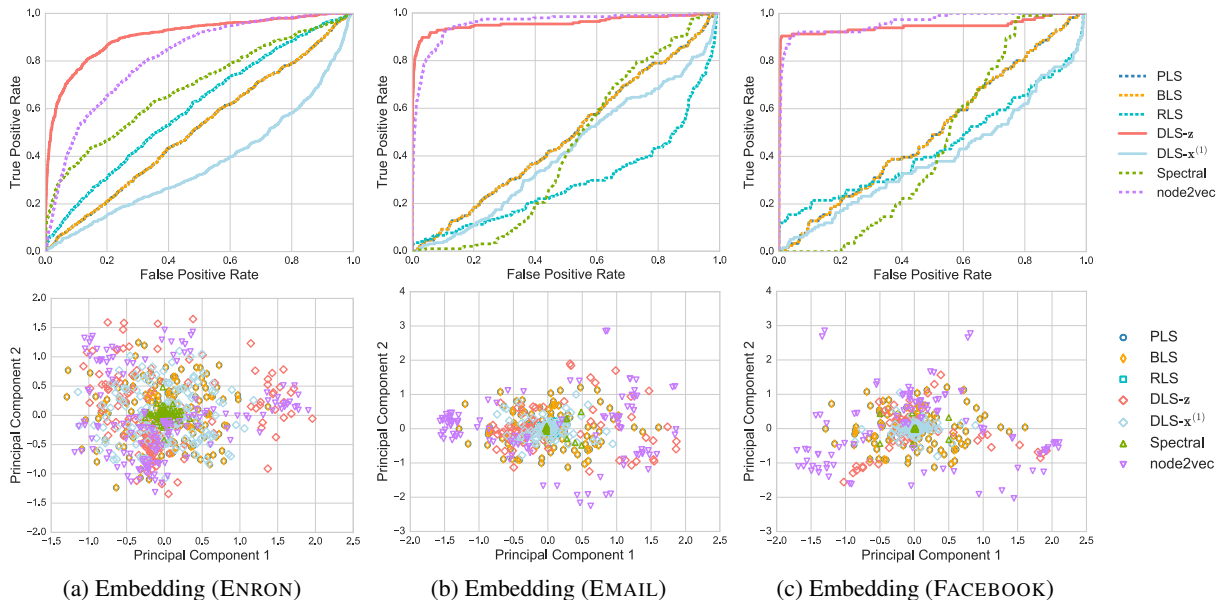(a) Embedding (ENRON) (b) Embedding (EMAIL) (c) Embedding (FACEBOOK)

Figure 1: Link prediction ROC curves (top row) and visualization of the learned embeddings (bottom row).

we have removed the reciprocal latent spaces, and the BLS results show that in that case the learned homophily latent space performs quite poorly in link prediction as well. This indicates that the reciprocal latent spaces may have a *denoising* effect—*i.e.*, that it "explains away" communications primarily due to reciprocity such that the remaining communications arising from intensities with low reciprocal component has to be due to the fact that the pair of nodes are inherently similar in some way, which is modeled by the homophily latent features.

We further visualize the estimated homophily and reciprocal latent spaces of the DLS model by computing the pair-wise similarities $e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$ for every pair of nodes $u, v \in V$, and then plotting a heat-map of the inferred similarity matrices. For the ENRON dataset, Figure 2 shows the heat-maps (colors on log-scale) for both the homophily latent space and the reciprocal latent space corresponding to an hourly exponential kernel (ϕ_1).[7] For each similarity matrix, we performed hierarchical clustering on the rows to obtain a node-ordering and accordingly permuted the rows and columns of the matrix simultaneously. Notice that the similarity matrices exhibit distinct clustering block-structures, indicating that the user-interaction patterns are quite different across the homophily and reciprocal latent spaces.

5.4 EXPLORING RECIPROCATION PATTERNS

While the reciprocal latent spaces in the DLS model may not be directly useful in static link prediction, they do

---

[7]The complete set of heat-maps for the remaining reciprocal latent spaces as well as those for EMAIL and FACEBOOK are provided in the supplementary material.

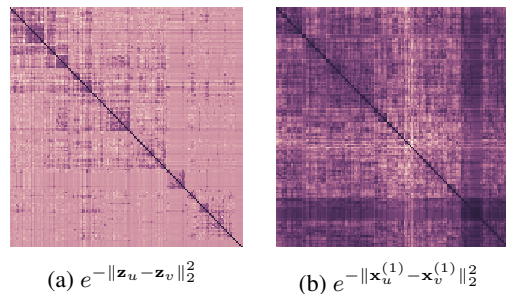(a) $e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$ (b) $e^{-\|\mathbf{x}_u^{(1)} - \mathbf{x}_v^{(1)}\|_2^2}$

Figure 2: Inferred node-similarity matrices in ENRON.

offer a unique tool for examining the varying reciprocation patterns exhibited across different triggering kernels. Specifically, for each pair of nodes u and v, we can compute their *relative* similarities in the b-th kernel via

$$p_{uv}^{(b)} \triangleq \frac{e^{-\|\mathbf{x}_u^{(b)} - \mathbf{x}_v^{(b)}\|_2^2}}{\sum_{h=1}^{B} e^{-\|\mathbf{x}_u^{(h)} - \mathbf{x}_v^{(h)}\|_2^2}}, \quad b = 1, \ldots, B.$$

This allows us to embed each pair of nodes onto a probability simplex where each pair $u, v \in V$ is represented by a point $(p_{uv}^{(1)}, \ldots, p_{uv}^{(B)})^{\mathsf{T}}$. Note that this simplicial embedding is of a different nature than the latent spaces themselves—if two points are nearby on this simplex, it indicates that the two pairs of nodes exhibit similar relative behavior across the chosen kernels, regardless of the absolute intensities of their communications.

In Figure 3, we selected two nodes in the ENRON network, and for each node v we plot the simplicial embeddings of each pair (v, u), $\forall u \in V$.[8] Figure 3 also plots

---

[8]For visualization, we have collapsed the kernels ϕ_3 and ϕ_4

node v's total outgoing intensity $\lambda_v(t) \triangleq \sum_{w \in V} \lambda_{vw}(t)$ as well as histograms showing the distribution of the Euclidean distances between v and the remaining nodes in the network. We observe that the two employees exhibit different reciprocation patterns with other employees in the corporation in terms of their active triggering kernels. For instance, the employee shown on the left appears to reciprocate with other employees in much of a similar manner since the points are more tightly concentrated, while the one on the right exhibits much more variability. Also notice that different reciprocating kernels may be active at different time-points, motivating the need for a mixture of kernel functions in modeling reciprocity.
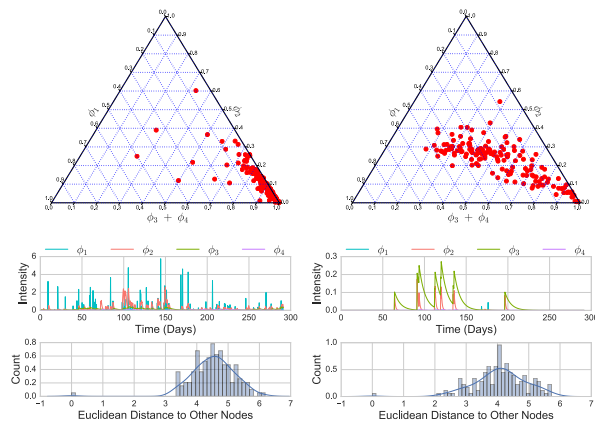

Figure 3: Visualizing reciprocation patterns in ENRON.

6 RELATED WORK

Point Processes. Recent work on point process models of structured temporal data include Simma and Jordan (2010); Perry and Wolfe (2013); DuBois et al. (2013); Guo et al. (2015); He et al. (2015); Farajtabar et al. (2015); Du et al. (2016); Tan et al. (2016). In Blundell et al. (2012), Hawkes processes were combined with the infinite relational model (Kemp et al., 2006) to perform nonparametric clustering of nodes. This forms a simplification to our models, with each node having a latent cluster index rather than a latent embedding. In this model, messages are observed by all nodes in a cluster rather than individual nodes, so that reciprocity operates at the cluster level. Blundell et al. (2012) also do not model heterogeneity in the reciprocating dynamics among users.

In Linderman and Adams (2014), the authors develop a framework that combines random graph priors on the latent network structure with a reciprocating point process observation model. This is roughly equivalent to our RLS model, which we use as a proxy for comparison to Linderman and Adams (2014). However, our focus is not on learning a latent network structure as much as on

_____

onto the same axis since they both have length-scale one week.

teasing apart complementary parts of an observed point process. Similar to Figure 2, our latent embeddings can be summarized with an associated graph; in this sense the DLS model can be thought to learn two complementary graph structures underlying events on a network.

Graph Embedding. Recent work in the graph mining community on learning feature representations for nodes in static networks include Perozzi et al. (2014); Tang et al. (2015); Grover and Leskovec (2016). The state-of-the-art approach is node2vec (Grover and Leskovec, 2016), which extends the skip-gram neural network architecture (Mikolov et al., 2013). Our experiments showed that by modeling both homophily and reciprocity in temporal interactions, the DLS model performs comparably or superior to node2vec in static link prediction.

7 CONCLUDING REMARKS

We have proposed latent space models for dynamic network data that embed the network nodes into Euclidean space. Our approach models heterogeneity across two important characteristics of such data—homophily and reciprocity—and connects latent space models of static networks to point process models including Poisson and Hawkes processes. The performance of our proposed dual latent space model shows that it is crucial to account for both characteristics to accurately model dynamic networks. In dynamic link prediction, we find that while the reciprocal latent space is important for accurate predictions, the inclusion of the homophily latent space produces a significant gain across all three real-world datasets. In static link prediction, while the reciprocal latent spaces are not directly useful for prediction, they greatly improve the quality of the estimated homophily latent space, by providing a denoising effect that filters out communications driven primarily by reciprocity.

Our findings shed further light on recent observations in Rudolph et al. (2016), who argue that modeling each observation *conditioned* on a set of other observations improves the quality of the learned embeddings. They refer to the conditioning set as *context* (*e.g.* in natural language the context of a word is its surrounding words). Similarly, one might argue the context of a node in a network is its neighbors. Including reciprocal latent spaces in the model implicitly conditions on the set of reciprocating neighbors, and including homophily latent spaces implicitly conditions on the set of similar neighbors.

References

A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

C. Blundell, K. Heller, and J. Beck. Modelling reciprocating relationships with Hawkes processes. In *NIPS*, 2012.

R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995.

N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD*, 2016.

C. DuBois, C. Butts, and P. Smyth. Stochastic block-modeling of relational event dynamics. In *AISTATS*, 2013.

D. Durante and D. Dunson. Nonparametric bayes dynamic modelling of relational data. *Biometrika*, 101 (4):883, 2014.

P. Ekeh. *Social exchange theory: The two traditions*. Heinemann London, 1974.

P. Erdős and A. Rényi. On random graphs, I. *Publicationes Mathematicae*, 6:290–297, 1959.

Y. Fan and C. R. Shelton. Learning continuous-time social network dynamics. In *UAI*, 2009.

M. Farajtabar, Y. Wang, M. Gomez-Rodriguez, S. Li, H. Zha, and L. Song. COEVOLVE: A joint point process model for information diffusion and network co-evolution. In *NIPS*, 2015.

W. Fu and E. Xing. Dynamic mixed membership block-model for evolving networks. In *ICML*, 2009.

A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, 2016.

F. Guo, C. Blundell, H. Wallach, and K. Heller. The Bayesian echo chamber: Modeling social influence via linguistic accommodation. In *AISTATS*, 2015.

S. Hanneke and E. Xing. Discrete temporal models of social networks. *Electron. J. Stat.*, 4:585–605, 2010.

X. He, T. Rekatsinas, J. Foulds, L. Getoor, and Y. Liu. Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In *ICML*, 2015.

P. Hoff, A. Raftery, and M. Handcock. Latent space approaches to social network analysis. *J. Am. Stat. Assoc.*, 97(460):1090–1098, 2002.

P. Holland and S. Leinhardt. An exponential family of probability distributions for directed graphs. *J. Am. Stat. Assoc.*, 76:33–50, 1981.

T. Iwata, A. Shah, and Z. Ghahramani. Discovering latent influence in online social activities via shared cascade Poisson processes. In *KDD*, 2013.

C. Kemp, J. Tenenbaum, and T. Griffiths. Learning systems of concepts with an infinite relational model. In *AAAI*, 2006.

B. Klimmt and Y. Yang. Introducing the enron corpus. In *CEAS*, 2004.

S. Linderman and R. Adams. Discovering latent network structure in point process data. In *ICML*, 2014.

M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annu. Rev. Sociol.*, 27(1):415–444, 2001.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

K. Miller, T. Griffiths, and M. Jordan. Nonparametric latent feature models for link prediction. In *NIPS*, 2009.

B. Min, K. Goh, and A. Vazquez. Spreading dynamics following bursty human activity patterns. *Phys. Rev. E*, 83(3):036102, 2011.

K. Nowicki and T. Snijders. Estimation and prediction for stochastic blockstructures. *J. Am. Stat. Assoc.*, 96: 1077–1087, 2001.

B. Perozzi, R. Al-Rfou, and S. Skiena. DeepWalk: Online learning of social representations. In *KDD*, 2014.

P. Perry and P. Wolfe. Point process modelling for directed interaction networks. *J. R. Stat. Soc. Ser. B*, 2013.

M. Rudolph, F. Ruiz, S. Mandt, and D. Blei. Exponential family embeddings. In *NIPS*, 2016.

P. Sarkar and A. Moore. Dynamic social network analysis using latent space models. In *NIPS*, 2005.

A. Simma and M. Jordan. Modeling events with cascades of Poisson processes. In *UAI*, 2010.

T. Snijders, G. van de Bunt, and C. Steglich. Introduction to stochastic actor-based models for network dynamics. *Social Networks*, 32(1):44–60, 2010.

X. Tan, S. Naqvi, A. Qi, K. Heller, and V. Rao. Content-based modeling of reciprocal relationships using Hawkes and Gaussian processes. In *UAI*, 2016.

J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. LINE: Large-scale information network embedding. In *WWW*, 2015.

U. von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17(4):395–416, 2007.

D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–42, 1998.

S. Young and E. Scheinerman. Random dot product graph models for social networks. In *WAW*, 2007.

A SUPPLEMENTARY MATERIAL

A.1 MAP ESTIMATION DETAILS

As described in Section 4.3, we perform maximum a posteriori (MAP) inference to estimate the parameters in all the discussed models. In this section, we present the MAP estimation details for the HP and DLS models by deriving the closed form expressions of the log-posterior function and its gradients; the optimization can then be carried out using L-BFGS-B (Byrd et al., 1995). The derivations for the PLS, BLS, RLS models follow analogously, since they can all be viewed as degenerate cases of the DLS model.

Before presenting the MAP estimation details, recall that the observed data $\{(u, v, \mathcal{H}_{uv})\}_{u,v \in V}$ are collected over a time period $[0, T)$, where $\mathcal{H}_{uv} \triangleq \{t_i^{uv}\}_{i=1}^{n_{uv}}$ records the set of all time-points at which u sent v a message.

A.1.1 Hawkes Process (HP) Model

Recall the Hawkes Process (HP) model:

$$\lambda_{uv}(t) = \gamma + \sum_{k:\, t_k^{vu} < t} \sum_{b=1}^{B} \xi_b\, \phi_b(t - t_k^{vu}) \qquad\qquad \forall u \neq v$$

$$N_{uv}(\cdot) \sim \mathsf{HawkesProcess}(\lambda_{uv}(\cdot)) \qquad\qquad \forall u \neq v$$

Notice that

$$\Lambda_{uv}(0, T) = \int_0^T \lambda_{uv}(t)\, \mathrm{d}t = \gamma\, T + \sum_{b=1}^{B} \xi_b \sum_{k=1}^{n_{vu}} [\Phi_b(T - t_k^{vu}) - \Phi_b(0)]$$

where $\Phi_b(t) \triangleq \int_0^t \phi_b(s)\, \mathrm{d}s$.

Placing $\mathsf{Gamma}(1, 1)$ priors on γ and each ξ_b, and denoting $\boldsymbol{\xi} \triangleq \{\xi_b\}_{b=1}^{B}$, the joint density can be written as

$$p(\{\mathcal{H}_{uv}\}_{u,v=1}^{n}, \gamma, \boldsymbol{\xi}) \propto \prod_{\substack{u,v=1 \\ u \neq v}}^{n} \left\{ e^{-\Lambda_{uv}(0,T)} \prod_{k=1}^{n_{uv}} \lambda_{uv}(t_i^{uv}) \cdot e^{-\gamma} \cdot \prod_{b=1}^{B} e^{-\xi_b} \right\}$$

and the log-posterior function is given by

$$\log p(\gamma, \boldsymbol{\xi} \,|\, \{\mathcal{H}_{uv}\}_{u,v=1}^{n}) = \sum_{\substack{u,v=1 \\ u \neq v}}^{n} \left\{ -\Lambda_{uv}(0, T) + \sum_{i=1}^{n_{uv}} \log \lambda_{uv}(t_i^{uv}) \right\} - \gamma - \sum_{b=1}^{B} \xi_b$$

$$= \sum_{\substack{u,v=1 \\ u \neq v}}^{n} \left\{ -\gamma\, T - \sum_{b=1}^{B} \xi_b\, \Delta_{b,T}^{vu} + \sum_{i=1}^{n_{uv}} \log \left(\gamma + \sum_{b=1}^{B} \xi_b\, \delta_{b,i}^{uv} \right) \right\} - \gamma - \sum_{b=1}^{B} \xi_b$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner-product, and we have adopted the shorthand notations

$$\Delta_{b,T}^{vu} \triangleq \sum_{k=1}^{n_{vu}} [\Phi_b(T - t_k^{vu}) - \Phi_b(0)]$$

$$\delta_{b,i}^{uv} \triangleq \sum_{k:\, t_k^{vu} < t_i^{uv}} \phi_b(t_i^{uv} - t_k^{vu})$$

to denote data statistics that can be pre-computed and cached for each pair of nodes $u, v \in V$ and kernel ϕ_b.

The gradients of the log-posterior are given by

$$\frac{\partial \log p}{\partial \gamma} = -\left(n^2 - n\right)T + \sum_{\substack{u,v=1 \\ u \neq v}}^{n} \sum_{i=1}^{n_{uv}} \left(\gamma + \sum_{b=1}^{B} \xi_b\, \delta_{b,i}^{uv}\right)^{-1} - 1$$

$$\frac{\partial \log p}{\partial \xi_b} = \sum_{\substack{u,v=1 \\ u \neq v}}^{n} \left[-\Delta_{b,T}^{vu} + \sum_{i=1}^{n_{uv}} \delta_{b,i}^{uv} \left(\gamma + \sum_{b=1}^{B} \xi_b\, \delta_{b,i}^{uv}\right)^{-1}\right] - 1\,.$$

A.1.2 Hawkes Dual Latent Space (DLS) Model

Recall the Hawkes Dual Latent Space (DLS) model:

$$\mathbf{z}_v \sim \mathcal{N}(\mathbf{0}, \sigma^2\, \mathbf{I}_{d\times d}) \qquad\qquad\qquad \forall v \in V$$
$$\boldsymbol{\mu}_v \sim \mathcal{N}(\mathbf{0}, \sigma_\mu^2\, \mathbf{I}_{d\times d}) \qquad\qquad\qquad \forall v \in V$$
$$\boldsymbol{\varepsilon}_v^{(b)} \sim \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2\, \mathbf{I}_{d\times d}) \qquad\qquad\qquad \forall v \in V,\ b = 1,\ldots,B$$
$$\mathbf{x}_v^{(b)} \sim \boldsymbol{\mu}_v + \boldsymbol{\varepsilon}_v^{(b)} \qquad\qquad\qquad \forall v \in V,\ b = 1,\ldots,B$$
$$\lambda_{uv}(t) = \gamma\, e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2} + \sum_{k:\, t_k^{vu} < t} \sum_{b=1}^{B} \beta\, e^{-\|\mathbf{x}_u^{(b)} - \mathbf{x}_v^{(b)}\|_2^2}\, \phi_b(t - t_k^{vu})$$
$$N_{uv}(\cdot) \sim \mathsf{HawkesProcess}(\lambda_{uv}(\cdot)) \qquad\qquad\qquad \forall u \neq v$$

Placing $\mathsf{Gamma}(1,1)$ priors on γ and β, setting $\sigma^2 = \sigma_\mu^2 = \sigma_\varepsilon^2 = 1$, and integrating out $\{\boldsymbol{\mu}_v\}_{v=1}^{n}$, the log-density function can be written as

$$\log p(\gamma, \beta, \{\mathbf{z}_v\}_{v=1}^{n}, \{\{\mathbf{x}_v^{(b)}\}_{b=1}^{B}\}_{v=1}^{n} \,|\, \{\mathcal{H}_{uv}\}_{u,v=1}^{n})$$
$$= \sum_{\substack{u,v=1 \\ u \neq v}}^{n} \left\{ -\gamma\, e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}\, T - \beta \sum_{b=1}^{B} \Delta_{b,T}^{vu}\, e^{-\|\mathbf{x}_u^{(b)} - \mathbf{x}_v^{(b)}\|_2^2} + \sum_{i=1}^{n_{uv}} \log\left(\gamma\, e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2} + \beta \sum_{b=1}^{B} \delta_{b,i}^{uv}\, e^{-\|\mathbf{x}_u^{(b)} - \mathbf{x}_v^{(b)}\|_2^2}\right)\right\}$$
$$- \frac{1}{2} \sum_{v=1}^{n} \sum_{b=1}^{B} \|\mathbf{x}_v^{(b)}\|_2^2 + \frac{B^2}{2\,(B+1)} \sum_{v=1}^{n} \|\bar{\mathbf{x}}_v\|_2^2 - \frac{1}{2} \sum_{v=1}^{n} \|\mathbf{z}_v\|_2^2 - \gamma - \beta$$

where $\bar{\mathbf{x}}_v \triangleq \frac{1}{B} \sum_{b=1}^{B} \mathbf{x}_v^{(b)}$ denotes the mean latent position of node v across all basis-kernels.

The gradients of the log-posterior are given by

$$\frac{\partial \log p}{\partial \gamma} = \sum_{\substack{u,v=1 \\ u \neq v}}^{n} \left[-T\, e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2} + \sum_{i=1}^{n_{uv}} e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}\, h^{-1}(u,v,i)\right] - 1$$

$$\frac{\partial \log p}{\partial \beta} = \sum_{\substack{u,v=1 \\ u \neq v}}^{n} \sum_{b=1}^{B} r(u,v,b)\, e^{-\|\mathbf{x}_u^{(b)} - \mathbf{x}_v^{(b)}\|_2^2} - 1$$

$$\nabla_{\mathbf{z}_v} \log p = \sum_{\substack{u=1 \\ u \neq v}}^{n} \left\{ \gamma \left[-2\,T + \sum_{i=1}^{n_{uv}} \left(h^{-1}(u,v,i) + h^{-1}(v,u,i)\right)\right] e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2} \cdot 2\,(\mathbf{z}_u - \mathbf{z}_v)\right\} - \mathbf{z}_v$$

$$\nabla_{\mathbf{x}_v^{(b)}} \log p = \sum_{\substack{u=1 \\ u \neq v}}^{n} \left\{ \beta\, [r(u,v,b) + r(v,u,b)]\, e^{-\|\mathbf{x}_u^{(b)} - \mathbf{x}_v^{(b)}\|_2^2} \cdot 2\,(\mathbf{x}_v^{(b)} - \mathbf{x}_u^{(b)})\right\} - \mathbf{x}_v^{(b)} + \frac{B}{B+1} \cdot \bar{\mathbf{x}}_v$$

where

$$h(u, v, i) \triangleq \gamma \, e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2} + \beta \sum_{b=1}^{B} \delta_{b,i}^{uv} \, e^{-\|\mathbf{x}_u^{(b)} - \mathbf{x}_v^{(b)}\|_2^2}$$

$$r(u, v, b) \triangleq -\Delta_{b,T}^{vu} + \sum_{i=1}^{n_{uv}} \delta_{b,i}^{uv} \, h^{-1}(u, v, i) \,.$$

A.2 ADDITIONAL EXPERIMENT RESULTS

A.2.1 Further Experiment on Static Link Prediction

In Section 5.3, we noted that the experiment setup for the static link prediction task did not yield standard errors for the AUC scores reported in Table 3, since there was only one training/test split. To investigate the statistical significance of the results, we conducted a follow-up experiment.

For each dataset, we computed confidence intervals by performing six trials on subsets of the data. Specifically, in the i-th trial, we let the training set to contain all events during the period between the $\lceil \frac{i-1}{10} \rceil$-th and the $\lfloor \frac{i+2}{10} \rfloor$-th event, and the test set to contain all events during the period between the $\lceil \frac{i+2}{10} \rceil$-th and $\lfloor \frac{i+4}{10} \rfloor$-th event. In this way, each trial used 30% training data and 20% test data, with the training and test data being non-overlapping.[9] As in Section 5.3, we fitted the model on the training set, and performed link prediction on the test set. The results are shown in Table 4.[10]

Table 4: Static link prediction AUC scores and standard deviations.

Model	ENRON	EMAIL	FACEBOOK
PLS	0.510 (0.009)	0.496 (0.015)	0.491 (0.013)
BLS	0.510 (0.009)	0.496 (0.015)	0.491 (0.013)
RLS	0.439 (0.073)	0.386 (0.081)	0.456 (0.055)
DLS	**0.864 (0.016)**	**0.934 (0.016)**	**0.892 (0.040)**
Spectral	0.516 (0.020)	0.526 (0.032)	0.492 (0.021)
node2vec	0.749 (0.050)	**0.953 (0.007)**	**0.935 (0.033)**

By conducting two-sided t-tests at the 95% confidence level, we conclude that while DLS significantly outperforms node2vec on ENRON, their performance differences on EMAIL and FACEBOOK are not significant.

A.2.2 Visualization of the Inferred Node-Similarity Matrices

We visualize the estimated homophily and reciprocal latent spaces of the DLS model by computing the pair-wise similarities $e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$ for every pair of nodes $u, v \in V$, and then plotting a heat-map of the inferred similarity matrices. Figures 4, 5, and 6 show the heat-maps (colors on log-scale) for both the homophily latent space and the reciprocal latent spaces corresponding to the hourly (ϕ_1), daily (ϕ_2), weekly (ϕ_3) exponential kernels and the weekly locally periodic kernel (ϕ_4) on all three datasets. For each similarity matrix, we performed hierarchical clustering on the rows to obtain a node-ordering and accordingly permuted the rows and columns of the matrix simultaneously. Notice that the similarity matrices exhibit different clustering block-structures, indicating that the user-interaction patterns are quite different across the homophily and reciprocal latent spaces with different kernels and time-scales.

---

[9]Notice, however, that the training/test data across different trials may share common observations. Thus, strictly speaking, the trials are not independent, and the computed standard error estimates might under-estimate the "true" associated uncertainty.

[10]Note that the overall performance for all methods are slightly degraded since we are only using subsets of the data.
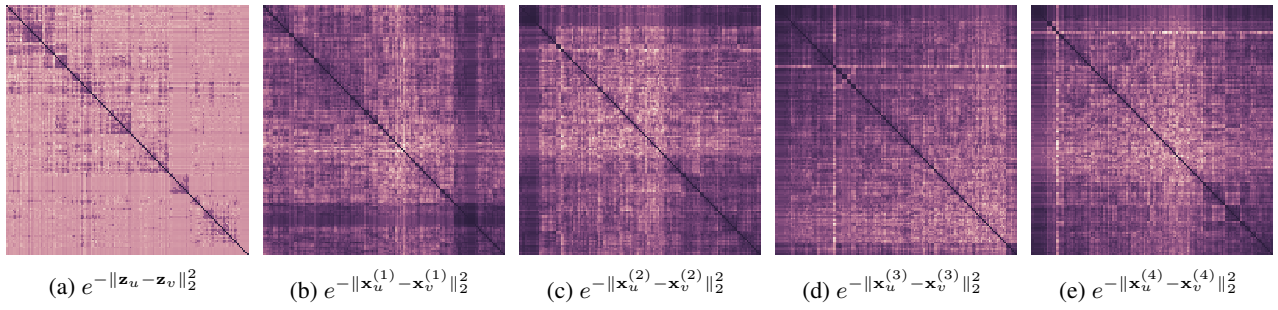
(a) $e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$ (b) $e^{-\|\mathbf{x}_u^{(1)} - \mathbf{x}_v^{(1)}\|_2^2}$ (c) $e^{-\|\mathbf{x}_u^{(2)} - \mathbf{x}_v^{(2)}\|_2^2}$ (d) $e^{-\|\mathbf{x}_u^{(3)} - \mathbf{x}_v^{(3)}\|_2^2}$ (e) $e^{-\|\mathbf{x}_u^{(4)} - \mathbf{x}_v^{(4)}\|_2^2}$

Figure 4: Inferred node-similarity matrices in ENRON.

(a) $e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$ (b) $e^{-\|\mathbf{x}_u^{(1)} - \mathbf{x}_v^{(1)}\|_2^2}$ (c) $e^{-\|\mathbf{x}_u^{(2)} - \mathbf{x}_v^{(2)}\|_2^2}$ (d) $e^{-\|\mathbf{x}_u^{(3)} - \mathbf{x}_v^{(3)}\|_2^2}$ (e) $e^{-\|\mathbf{x}_u^{(4)} - \mathbf{x}_v^{(4)}\|_2^2}$
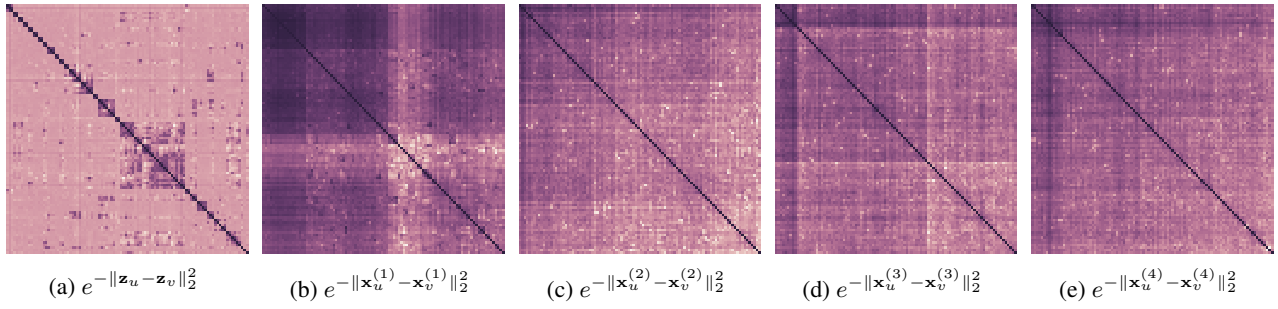
Figure 5: Inferred node-similarity matrices in EMAIL.

(a) $e^{-\|\mathbf{z}_u - \mathbf{z}_v\|_2^2}$ (b) $e^{-\|\mathbf{x}_u^{(1)} - \mathbf{x}_v^{(1)}\|_2^2}$ (c) $e^{-\|\mathbf{x}_u^{(2)} - \mathbf{x}_v^{(2)}\|_2^2}$ (d) $e^{-\|\mathbf{x}_u^{(3)} - \mathbf{x}_v^{(3)}\|_2^2}$ (e) $e^{-\|\mathbf{x}_u^{(4)} - \mathbf{x}_v^{(4)}\|_2^2}$
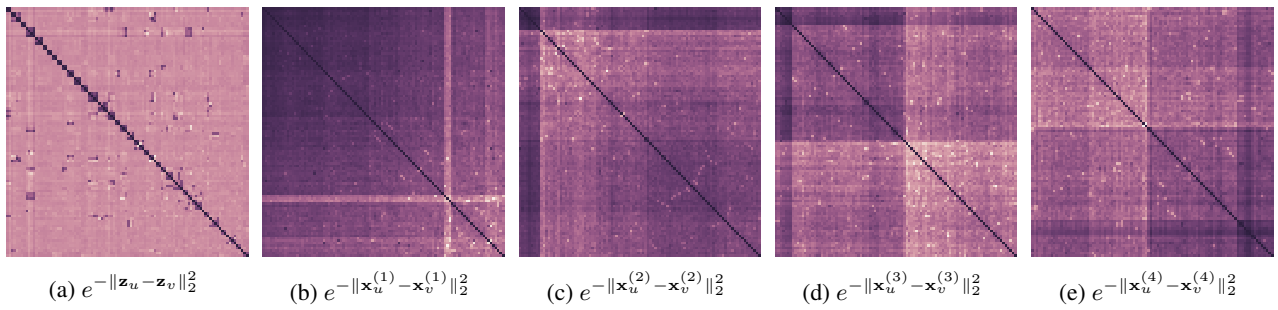
Figure 6: Inferred node-similarity matrices in FACEBOOK.