Coupled Point Process-based Sequence Modeling for Privacy-preserving Network Alignment

Dixin Luo[1⋆], **Haoran Cheng**[1⋆], **Qingbin Li**[1], **Hongteng Xu**[2,3]
[1]School of Computer Science and Technology, Beijing Institue of Technology
[2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
{dixin.luo, haoran.cheng, 1120200298}@bit.edu.cn, hongtengxu@ruc.edu.cn

Abstract

Network alignment aims at finding the correspondence of nodes across different networks, which is significant for many applications, e.g., fraud detection and crime network tracing across platforms. In practice, however, accessing the topological information of different networks is often restricted and even forbidden, considering privacy and security issues. Instead, what we observed might be the event sequences of the networks' nodes in the continuous-time domain. In this study, we develop a coupled neural point process-based (CPP) sequence modeling strategy, which provides a solution to privacy-preserving network alignment based on the event sequences. Our CPP consists of a coupled node embedding layer and a neural point process module. The coupled node embedding layer embeds one network's nodes and explicitly models the alignment matrix between the two networks. Accordingly, it parameterizes the node embeddings of the other network by the push-forward operation. Given the node embeddings, the neural point process module jointly captures the dynamics of the two networks' event sequences. We learn the CPP model in a maximum likelihood estimation framework with an inverse optimal transport (IOT) regularizer. Experiments show that our CPP is compatible with various point process backbones and is robust to the model misspecification issue, which achieves encouraging performance on network alignment. The code is available at https://github.com/Dixin-s-Lab/CNPP.

1 Introduction

As a fundamental task for network modeling and analysis, network alignment plays a central role in many real-world applications, such as financial fraud detection [Zhang *et al.*, 2019; Pourhabibi *et al.*, 2020] and crime network tracing [Wang *et al.*, 2019a; Sun *et al.*, 2022] across different platforms, and is significant for social good. Given two or more networks, we often model network alignment as a graph matching problem and infer the correspondence between the networks based on their topologies (e.g., the adjacency matrices indicating edges [Bayati *et al.*, 2009; Mohammadi *et al.*, 2017] and the similarity matrices derived from node features [Heimann *et al.*, 2018; Xu *et al.*, 2019]). Following this modeling strategy, many network alignment methods have been proposed. However, these methods ignore that the network topology is unreliable and even unavailable in many real-world scenarios, which limits their applications.

Take financial crime network identification and tracing as an example. Criminals often have multiple accounts across different financial platforms with fake identities. Detecting and tracing crime networks across different platforms needs to access the network topologies of multiple platforms and align the networks based on their topological similarities. However, for the service providers of the platforms, sharing their network topologies with others raises the risk of customer information leakage because the network topology often contains the private information of normal accounts (e.g., their identities, profiles, and social connections) besides the criminals' accounts. As a result, coordinating with multiple platforms to obtain their network topologies is often technically and politically infeasible for a third party. Such a scenario leads to a significant and challenging *privacy-preserving network alignment problem*, which requires us to align networks without accessing their topologies.

The privacy-preserving network alignment problem is highly correlated with the 16th United Nations Sustainable Development Goal (UN-SDG), i.e., promoting peaceful and inclusive societies for sustainable development, providing access to justice for all, and building effective, accountable, and inclusive institutions at all levels. On the one hand, network alignment is one of the necessary techniques for combating global cybercrimes, especially for detecting and tracing financial fraud and money laundering across platforms. On the other hand, privacy preservation is an objective restriction for any institution accessing multi-network data, which helps to protect civil rights. While this problem commonly appears in various practical applications, to our surprise, it is seldom considered by existing network alignment works.

In this study, we develop a novel coupled point process (CPP) model associated with a robust learning algorithm, which provides a potential solution to privacy-preserving network alignment. As illustrated in Figure 1, our method aligns two networks based on the event sequences of their nodes in the continuous-time domain rather than the network topolo-

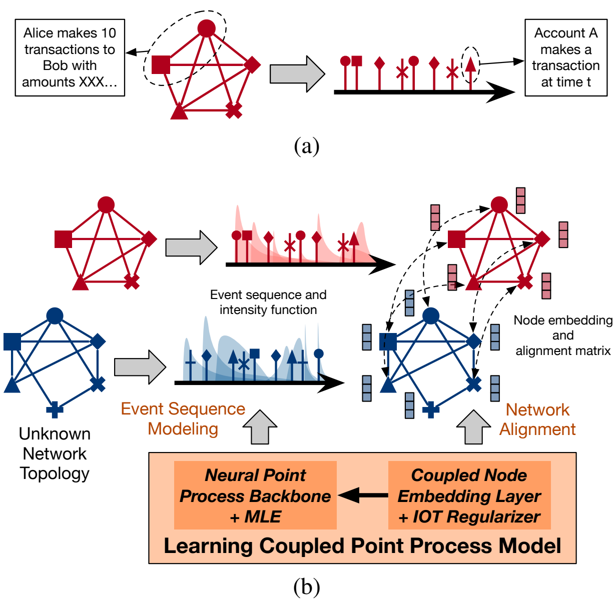


Figure 1: (a) The network topology v.s. Anonymous event sequence of nodes. As shown in the boxes, the event sequence often contains less private information. (b) The scheme of our CPP model.

gies. This setting is based on two facts: i) The sequential behaviors of nodes are often driven by their social relations (i.e., the network topology) and thus contain important information for our alignment task (i.e., the nodes having correspondence should have similar sequential behaviors). ii) Compared to the network topology, the sequential behaviors of nodes are relatively easy to access and with much less private information.[1] Our CPP model consists of two modules: i) a coupled node embedding layer capturing the node embeddings specified for different networks, and ii) a shared neural point process module predicting the dynamics of the event sequences jointly. Instead of embedding the nodes independently, the coupled embedding layer explicitly models one network's node embeddings and parameterizes the other network's node embeddings by multiplying the explicit node embeddings with a learnable alignment matrix. The alignment matrix is doubly-stochastic, which indicates the correspondence between the two networks' nodes in a probabilistic manner. The CPP model can be efficiently learned by the maximum likelihood estimation with an inverse optimal transport (IOT) regularizer [Li *et al.*, 2019].

To our knowledge, our work makes the first attempt to achieve privacy-preserving network alignment based on a coupled point process-based modeling strategy. For our CPP model, its coupled node embedding layer provides a new technical route to jointly model the node embeddings of different networks. In the learning phase, the IOT-based regularizer works better than other candidate regularizers, which leads to robust alignment results. Experiments show that our method is compatible with various point process backbones [Zuo *et al.*, 2020; Zhang *et al.*, 2020a] and thus is robust to the model misspecification issue. It outperforms other sequence-based alignment methods [Luo *et al.*, 2019] and is comparable to the traditional methods that rely on network topology, which achieves a trade-off between alignment accuracy and privacy preservation.

[1]For example, for a criminal with a financial account, accessing the transaction history of the account is easier than inferring the identity and the social network of the criminal [Zhu and Xie, 2022].

2 Related Work

2.1 Network Alignment

Traditional network alignment methods can be categorized into two classes. One is extracting manually-designed node features and aligning them via heuristic algorithms, e.g., the genetic algorithms used in [Sun *et al.*, 2015; Vijayan *et al.*, 2015], the greedy search in [Neyshabur *et al.*, 2013; Huang *et al.*, 2016], and the spectral methods in [Patro and Kingsford, 2012; Nassar *et al.*, 2018]. The other formulates the task as a quadratic assignment problem (QAP) and solves it approximately under different relaxation strategies and with various structural information. The commonly-used relaxation strategies include the convex relaxation [Hashemifar *et al.*, 2016] and the doubly-stochastic relaxation based on the optimal transport theory [Xu *et al.*, 2019]. For the structural information, besides considering the pairwise relations [Liu *et al.*, 2016], the TAME in [Mohammadi *et al.*, 2017] further considers high-order relations among nodes when aligning networks. These methods are based on the consistency assumption, i.e., the two nodes of different networks are likely to be aligned if their neighborhoods have similar topologies.

Recently, some learning-based network alignment methods have been proposed, which align networks based on learned node embeddings. Typical node embedding strategies include the matrix factorization used in REGAL [Heimann *et al.*, 2018], the random walk-based method in BRIGHT [Yan *et al.*, 2021], and the graph convolution network in NeXtAlign [Zhang *et al.*, 2021]. These methods apply learnable modules to embed nodes and align the networks based on the similarity matrix constructed by the node embeddings. More recently, beyond learning node embeddings, some attempts have been made to make the whole network alignment task learnable in an end-to-end manner, e.g., the graph matching network (GMN) in [Zanfir and Sminchisescu, 2018], the PIA/PCA-GM in [Wang *et al.*, 2019b], and the DGMC in [Fey *et al.*, 2020]. These methods learn a differentiable alignment matrix for network alignment via the Sinkhorn matching module [Adams and Zemel, 2011].

2.2 Point Process-Based Network Modeling

Given the event sequences generated based on network topology, we often apply temporal point processes (TPPs) [Daley and Vere-Jones, 2007] to capture their dynamics. One of the most well-known TPPs is the Hawkes process [Hawkes, 1971], which can infer network topology directly by exploring the Granger causal graph of nodes from the event sequences [Zhou *et al.*, 2013; Xu *et al.*, 2016a]. Recently, some variants of Hawkes process have been proposed, e.g., RMTPP [Du *et al.*, 2016], CT-LSTM [Mei and Eisner, 2017], self-attentive Hawkes process (SAHP) [Zhang *et al.*, 2020a], and Transformer Hawkes process (THP) [Zuo *et al.*, 2020]. These models enhance the representation power of Hawkes process via neural networks, e.g., recurrent neural networks and attention layers [Vaswani *et al.*, 2017]. The work in [Zhang *et al.*, 2020c] shows that these neural point processes can learn the Granger causal graph of nodes (i.e., the network topology) as the traditional Hawkes process does. We often learn the TPP models based on the maximum likelihood estimation (MLE) [Daley and Vere-Jones, 2007]. The work in [Guo *et al.*, 2018; Mei *et al.*, 2020] applies contrastive learning strategies to speed up the learning process.

The above TPP models have been applied to many practical applications, e.g., financial data analysis [Bacry *et al.*, 2015] and crime network detection [Mohler *et al.*, 2011; Zhu and Xie, 2022]. However, given the event sequences generated from two or more networks, their joint modeling is seldom considered by existing work. Recently, the fused Gromov-Wasserstein alignment (FGWA) method is proposed for Hawkes processes [Luo *et al.*, 2019], which aligns networks based on the event sequences of nodes. This method is hard to expand to large-scale networks and sensitive to hyperparameter settings. Moreover, it assumes that the event sequences are generated by random processes predefined on the networks, resulting in a high model misspecification risk.

3 Proposed Model

3.1 Preliminaries and Problem Statement

Suppose that we have two networks, which are represented as two graphs, i.e., $\{G_k(\mathcal{V}_k, \boldsymbol{A}_k)\}_{k=1}^2$. For the k-th network, $\mathcal{V}_k = \{v_i^k\}_{i=1}^{I_k}$ is the node set, and $\boldsymbol{A}_k \in \mathbb{R}^{I_k \times I_k}$ is the adjacency matrix, whose non-zero elements indicates the edges. As aforementioned, we consider the scenario where neither the topological structures of the graphs (the $\boldsymbol{A}_k$'s) nor the correspondence among them is known. What we observed are the event sequences of their nodes in the continuous-time domain, denoted as $\mathcal{S}_k = \{(t_m^k, v_m^k) \in [0,T] \times \mathcal{V}_k\}_{m=1}^{M_k}$ for $k = 1, 2$. Here, $\mathcal{S}_k$ represents the event sequence generated by the k-th network, and (t_m^k, v_m^k) represents the m-th event of the node $v_m^k \in \mathcal{V}_k$ at the timestamp $t_m^k \in [0, T]$, where T indicates the length of the time window.

We assume that the event sequence of the k-th network is generated by an unknown temporal point process (TPP), i.e.,

$$\mathcal{S}_k \sim \boldsymbol{N}_k(t; \boldsymbol{A}_k), \quad \text{for } k = 1, 2, \tag{1}$$

where $\boldsymbol{N}_k(t; \boldsymbol{A}_k) = \{N_v^k(t; \boldsymbol{A}_k)\}_{v \in \mathcal{V}_k}$ is the counting process corresponding to the temporal point process, whose parameters are determined by the adjacency matrix $\boldsymbol{A}_k$. $N_v^k(t; \boldsymbol{A}_k)$ counts the number of the node v's events till time t, and we characterize the expected instantaneous happening rates of the events via an intensity function, i.e.,

$$\lambda_v^k(t)\mathrm{d}t = \mathbb{E}[\mathrm{d}N_v^k(t) | \mathcal{H}_t^k], \quad \text{for } v \in \mathcal{V}_k, \tag{2}$$

where v indicates the node index belonging to a node set $\mathcal{V}_k$, and $\mathcal{H}_t^k = \{(t_m^k, v_m^k) \in \mathcal{S}_k | t_m^k < t\}$ represents the historical events till time t.

Suppose that the two networks have correspondence with each other, i.e.,

$$\boldsymbol{A}_2 = \boldsymbol{P}\boldsymbol{A}_1\boldsymbol{P}^T + \boldsymbol{E}, \tag{3}$$

where the alignment matrix $\boldsymbol{P} \in \{0,1\}^{N_2 \times N_1}$ indicates the correspondence between the two networks' nodes, and $\boldsymbol{E} \in \mathbb{R}^{N_2 \times N_1}$ represents the unknown random noise. Given $\{\mathcal{S}_k\}_{k=1}^2$, we would like to infer $\boldsymbol{P}$ robustly.

This privacy-preserving network alignment problem is more challenging than the traditional one because it introduces more uncertainty. In particular, neither the adjacency matrices nor the alignment matrix is known, so we have to solve the network inference and alignment jointly. A naïve way is solving the following learning task:

$$\min_{\{\boldsymbol{A}_k\}_{k=1}^2, \boldsymbol{P}} \sum_{k=1}^2 \mathcal{L}(\mathcal{S}_k; \boldsymbol{A}_k) + \tau \mathrm{R}(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{P}), \tag{4}$$

where $\mathcal{L}(\mathcal{S}_k; \boldsymbol{A}_k)$ is the negative log-likelihood (NLL) of the sequence $\mathcal{S}_k$, which is defined as

$$\sum_{v \in \mathcal{V}_k} \int_0^T \lambda_v^k(s)\mathrm{d}s - \sum_{m=1}^{M_k} \log \lambda_{v_m^k}^k(t_m^k), \tag{5}$$

and $\mathrm{R}(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{P})$ is a regularizer for the network topology and the alignment matrix, whose significance is controlled by $\tau > 0$. We can design the regularizer in various manners. For example, it can be $\|\boldsymbol{A}_2 - \boldsymbol{P}\boldsymbol{A}_1\boldsymbol{P}^T\|_F^2$ based on the model assumption in (3). In [Luo *et al.*, 2019], it is defined as a Gromov-Wasserstein (GW) alignment loss [Xu *et al.*, 2019], i.e., $\mathrm{R}(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{P}) = \max_{\boldsymbol{P} \in \Omega} \langle \boldsymbol{A}_2 \boldsymbol{P} \boldsymbol{A}_1, \boldsymbol{P} \rangle$, where Ω is a doubly-stochastic constraint on $\boldsymbol{P}$ and $\langle \cdot, \cdot \rangle$ indicates the inner product of matrices.

Unfortunately, this naïve method suffers from the following three problems. Firstly, learning the TPP models and the alignment matrix jointly has much higher complexity and often leads to unsatisfactory sub-optimal solutions. Secondly, the scalability of this learning method is poor because the complexity of the model parameters is quadratic to the number of nodes. Finally, the formulation of the TPP model in (1) is often unknown in practice, so the learning problem often suffers from the model misspecification issue. Based on the analysis above, we need to design a TPP model that has better scalability and is robust to the model misspecification problem. Additionally, we need to introduce prior knowledge and design a more informative regularizer to avoid catastrophic over-fitting. These requirements motivate us to design the coupled point process model and its learning algorithm.

3.2 Coupled Point Process Model

Given the event sequences of two networks, our CPP models their dynamics jointly based on the same conditional intensity function instead of modeling them independently. Specifically, given a timestamp t and the historical events $\mathcal{H}_t$ (from either $\mathcal{S}_1$ or $\mathcal{S}_2$), we have

$$\lambda_v(t) = g(t, \mathcal{H}_t; \{\boldsymbol{V}_k\}_{k=1}^2, \theta), \quad \text{for } v \in \mathcal{V}_1 \cup \mathcal{V}_2 \tag{6}$$

where $\boldsymbol{V}_k = [\boldsymbol{v}_i^k] \in \mathbb{R}^{D \times I_k}$ represents the node embeddings of the k-th network. g is a neural TPP that takes the timestamp and the history as input. Its parameters include: $i)$ the node embeddings $\{\boldsymbol{V}_k\}_{k=1}^2$ that are specified for each network, and $ii)$ the parameter θ shared by the two networks.

The CPP in (6) is a generalized framework covering many representative temporal point process models. In particular, we can implement g based on different backbones and obtain various point processes accordingly.

Classic Point Processes. Our CPP can cover classic TPPs, e.g., Hawkes processes [Zhou *et al.*, 2013] and mutually-correcting processes [Xu *et al.*, 2016b], via parameterizing the adjacency matrices based on the node embeddings. For the Hawkes process, denoted as HP($\boldsymbol{\mu}, \boldsymbol{\Phi}$), its intensity function $\lambda_v(t) = \mu_v + \sum_{t_n<t} \phi_{vv_n}(t - t_n)$ consists of a base intensity $\boldsymbol{\mu} = [\mu_v]$ and a set of impact functions $\boldsymbol{\Phi} = [\phi_{vv'}(t)]$. Each impact function is often modeled based on the adjacency matrix and a decay function, i.e., $\phi_{vv'}(t) = a_{vv'}\kappa(t)$, where $a_{vv'}$ is the element of $\boldsymbol{A}$. Our CPP parameterizes $\boldsymbol{A}$ as a bi-linear model $\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{V}$ and $\boldsymbol{\mu}$ as a linear model $\boldsymbol{V}^T\boldsymbol{w}$, and accordingly, $\theta = \{\boldsymbol{W} \in \mathbb{R}^{D\times D}, \boldsymbol{w} \in \mathbb{R}^D\}$.

Neural Hawkes Processes. When g is a multi-head self-attention module, our CPP corresponds to the Transformer Hawkes process (THP) [Zuo *et al.*, 2020] or the self-attentive Hawkes process (SAHP) [Zhang *et al.*, 2020a]. When g is a recurrent neural network module defined in the continuous-time domain, our CPP corresponds to the continuous-time LSTM (CT-LSTM) [Mei and Eisner, 2017] or the RMTPP [Du *et al.*, 2016]. These TPPs take learnable node embeddings as input.

Our CPP models the event sequences of two networks jointly, whose number of parameters is fewer than that of modeling the sequences independently. Additionally, instead of inferring the adjacency matrices, our CPP captures the topological information of the networks by the node embeddings. As a result, our model has better scalability and a lower risk of over-fitting because the number of model parameters is linear rather than quadratic to the network size.

Based on the node embeddings, we reformulate the network alignment problem as inferring an alignment matrix to indicate the similar node embeddings across the two networks, i.e., $\boldsymbol{V}_2 = \boldsymbol{V}_1\boldsymbol{P}^T$. The key point of our CPP model is the following coupled node embedding layer, which takes the alignment matrix as a part of the model parameters.

3.3 Coupled Node Embedding Layer

As illustrated in Figure 2, our coupled node embedding layer models the node embeddings of one network explicitly and parameterizes the node embeddings of the other network as

$$\boldsymbol{V}_2 := \boldsymbol{V}_1\boldsymbol{P}_\gamma^T, \tag{7}$$

where $\boldsymbol{P}_\gamma$ is an approximation of the alignment matrix $\boldsymbol{P}$, which is parameterized by γ. To make the layer differentiable, we relax the binary restriction of the alignment matrix, modeling $\boldsymbol{P}_\gamma$ by the following Sinkhorn matching module [Adams and Zemel, 2011]:

$$\begin{aligned}\boldsymbol{P}_\gamma &= I_2\text{Sinkhorn}(\boldsymbol{C}_\gamma, \epsilon)\\ &= I_2 \arg\min\nolimits_{\boldsymbol{T}\in\Pi(\frac{1}{I_2}\boldsymbol{1}_{I_2}, \frac{1}{I_1}\boldsymbol{1}_{I_1})} \langle \boldsymbol{C}_\gamma, \boldsymbol{T}\rangle + \epsilon\langle \boldsymbol{T}, \log \boldsymbol{T}\rangle,\end{aligned} \tag{8}$$

where Sinkhorn($\boldsymbol{C}_\gamma, \epsilon$) denotes the matching module based on the Sinkhorn scaling algorithm [Sinkhorn and Knopp, 1967; Cuturi, 2013], which solves the entropic optimal transport (EOT) problem in (8) via Algorithm 1. Here, $\boldsymbol{C}_\gamma \in \mathbb{R}^{I_2\times I_1}$ is the grounding cost matrix of the EOT problem, which is with parameter γ. In this study, we consider two implementations of $\boldsymbol{C}_\gamma$: i) treating it as a *non-parametric* model and learning the matrix directly; ii) further parameterizing it by a neural network (e.g., a *parametric* model like an MLP) and learning the neural network instead.

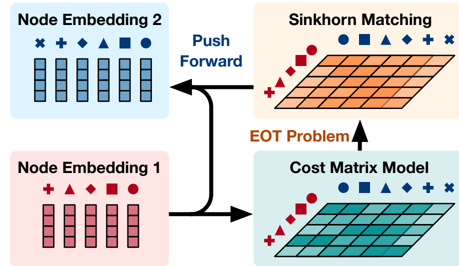


Figure 2: An illustration of the coupled node embedding layer.

Algorithm 1 Sinkhorn matching module

Input: The cost matrix $\boldsymbol{C}_\gamma$.
Hyperparameters: The number of iterations K, the weight of the entropic regularizer ϵ.
Initialize $\boldsymbol{K} = \exp(-\frac{\boldsymbol{C}_\gamma}{\epsilon})$, $\boldsymbol{a} = \boldsymbol{1}_{I_1}$, and $\boldsymbol{b} = \boldsymbol{1}_{I_2}$
for $k = 1, ..., K$ **do**
    $\boldsymbol{b} \leftarrow \frac{1}{I_2\boldsymbol{K}\boldsymbol{a}}$, then $\boldsymbol{a} \leftarrow \frac{1}{I_1\boldsymbol{K}^T\boldsymbol{b}}$.
end for
Output: $\boldsymbol{T}^* \leftarrow \text{diag}(\boldsymbol{b})\boldsymbol{K}\text{diag}(\boldsymbol{a})$.

The optimal solution of the EOT problem, denoted as $\boldsymbol{T}^*$ is a doubly-stochastic matrix, i.e., $\boldsymbol{T}^* \in \Pi(\frac{1}{I_2}\boldsymbol{1}_{I_2}, \frac{1}{I_1}\boldsymbol{1}_{I_1}) = \{\boldsymbol{T} \in \mathbb{R}_+^{I_2\times I_1} | \boldsymbol{T}\boldsymbol{1}_{I_1} = \frac{1}{I_2}\boldsymbol{1}_{I_2}, \boldsymbol{T}^T\boldsymbol{1}_{I_2} = \frac{1}{I_1}\boldsymbol{1}_{I_1}\}$. Therefore, $\boldsymbol{P}_\gamma = I_2\boldsymbol{T}^*$ is a transition matrix, which provides a probabilistic approximation of the alignment matrix $\boldsymbol{P}$. In particular, the element of $\boldsymbol{P}_\gamma$, denoted as $p(v|v'; \gamma)$, represents the conditional probability of $v \in \mathcal{V}_2$ given $v' \in \mathcal{V}_1$. Note that the proposed coupled node embedding layer can be easily extended to multi-network scenarios — when new networks come, we merely need to add more Sinkhorn matching modules to derive their node embeddings.

4 Learning Algorithm

We learn the CPP model via a regularized maximum likelihood framework. In this framework, the likelihood of the event sequences is maximized, and the alignment matrix is regularized by an intensity-based priori matrix in an inverse optimal transport (IOT) format.

4.1 Intensity-Based Prior of Alignment Matrix

As aforementioned, the event sequences generated by the networks often contain useful information for network alignment. Given the event sequences, the significance of a node can be measured by the average intensity of its events, i.e.,

$$\bar{\lambda}_v(T) = N_v(T)/T, \ \forall v \in \mathcal{V}_1 \cup \mathcal{V}_2. \tag{9}$$

The average intensity records the density of observed events in the time window $[0, T]$, which reflects the activity of the node v. We assume that the nodes with similar average intensity should be aligned with a higher probability. Based on this assumption, we can construct an intensity-based prior

Algorithm 2 Learning the proposed CPP model

1: **Input:** Event sequences $\{\mathcal{S}_k\}_{k=1}^{2}$.
2: **Hyperparameters:** The number of iterations M, the weight of the IOT regularizer τ.
3: Compute the average intensity for each node in $\mathcal{V}_1 \cup \mathcal{V}_2$ and construct $\boldsymbol{C}$.
4: Compute the priori alignment matrix $\boldsymbol{P}_0$ via Algorithm 1 and fix it as a constant.
5: **for** $m = 1, ..., M$ **do**
6: Compute the loss function in (10) and udpate the parameters by Adam [Kingma and Ba, 2014].
7: **end for**
8: **Output:** Learned alignment matrix $\boldsymbol{P}_{\gamma^*}$.

for the alignment matrix. In particular, we first calculate the average intensity for each node and construct a cost matrix $\boldsymbol{C} = [c_{vv'}] \in \mathbb{R}^{I_2 \times I_1}$, where $c_{vv'} = |\bar{\lambda}_v(T) - \bar{\lambda}_{v'}(T)|$ for $v \in \mathcal{V}_2$ and $v' \in \mathcal{V}_1$. Then, we reuse the Sinkhorn matching module in Algorithm 1, computing a priori alignment matrix, denoted as $\boldsymbol{P}_0 = I_2\text{Sinkhorn}(\boldsymbol{C}, \epsilon)$.

It should be noted that the assumption behind the prior is reasonable, which matches the principle of network alignment. In practice, given two networks, we often first align the key nodes that have high degrees and then deal with other nodes [Xu *et al.*, 2019; Malod-Dognin and Pržulj, 2015] because these key nodes have sufficient and distinguishable topological information. In our privacy-preserving scenario, although the network topology is unavailable, we can still detect the key nodes based on the average intensity in (9) — the key nodes, e.g., the leaders in a network, often have active behaviors (i.e., high average intensity) and thus have significant impacts on other nodes. Moreover, the average intensity is an unbiased estimation of $\mathbb{E}[\lambda(T)]$, i.e., $\bar{\lambda}_v(T) \to \mathbb{E}[\lambda(T)]$ with the increase of time and event number. In other words, the average intensity of the key nodes is more reliable than that of other nodes because of the sufficiency of events. Accordingly, the priori alignment matrix may provide useful evidence for the alignment of the key nodes.

4.2 Regularized Maximum Likelihood Estimation

Given the event sequences of two networks and the priori alignment matrix, we learn our CPP model via solving the following optimization problem:

$$\min_{\boldsymbol{V}_1, \boldsymbol{P}_\gamma, \theta} \mathcal{L}(\mathcal{S}_1 \cup \mathcal{S}_2; \boldsymbol{V}_1, \boldsymbol{P}_\gamma, \theta) + \tau \text{KL}(\boldsymbol{P}_\gamma \| \boldsymbol{P}_0), \quad (10)$$

where $\mathcal{S}_1 \cup \mathcal{S}_2$ is the collection of the event sequences of the two networks. They are modeled jointly by our CPP model, and $\mathcal{L}$ is the negative log-likelihood defined in (5). $\text{KL}(\boldsymbol{P}_\gamma \| \boldsymbol{P}_0)$ penalizes the KL-divergence between the approximated alignment matrix and its prior. From the viewpoint of optimal transport, $\frac{1}{I_2}\boldsymbol{P}_\gamma$ is a transport matrix derived based on a learnable cost matrix $\boldsymbol{C}_\gamma$, while $\frac{1}{I_2}\boldsymbol{P}_0$ is a known transport matrix. Accordingly, the KL-divergence leads to the inverse optimal transport problem [Li *et al.*, 2019], i.e., optimizing the cost matrix given a predefined transport matrix.

We solve (10) by stochastic gradient descent (SGD), as shown in Algorithm 2. Note that, when implementing our CPP via neural point processes, the integral in the negative log-likelihood is approximated by the Monte Carlo integration, as the work in [Mei and Eisner, 2017; Zhang *et al.*, 2020a; Zuo *et al.*, 2020] did.

Dataset		#Nodes	#Edges	#Sequences	Noise
SynData	G_1	10	21	2,000	No
($I = 10$)	G_2	10	21	2,000	
SynData	G_1	50	597	2,000	No
($I = 50$)	G_2	50	597	2,000	
SynData	G_1	100	2,548	2,000	No
($I = 100$)	G_2	100	2,548	2,000	
Arenas	G_1	1,133	5,452	10,000	Light
	G_2	1,133	5,150	10,000	
Cora	G_1	2,708	5,812	10,000	Medium
	G_2	2,708	4,445	10,000	
Phone-	G_1	1000	41,191	10,000	Heavy
Email	G_2	1003	4,627	10,000	

Table 1: Basic statistics of datasets.

5 Experiments

5.1 Experimental Setup

To demonstrate the feasibility and usefulness of our CPP model in privacy-preserving network alignment tasks, we conduct experiments on both synthetic and real-world datasets. The pairwise networks we considered include:

SynData contains three synthetic network pairs. In each pair, the source network G_1 is an Erdös-Rényi graph [Gilbert, 1959] with self-loops, and the target network G_2 is an isomorphism of G_1. Accordingly, the ground truth alignment matrix $\boldsymbol{P}$ corresponds to a permutation matrix. For the networks, we set the number of nodes $I \in \{10, 50, 100\}$.

Arenas is a network recording email communications among 1,133 users [Leskovec and Sosič, 2016]. Taking this network as G_1, we follow the network alignment literature [Koutra *et al.*, 2013; Zhang and Tong, 2016; Heimann *et al.*, 2018] and construct G_2 by removing the edges of G_1 with probability 0.05 (and without disconnecting any nodes).

Cora records a citation network among 2,708 publications [Yang *et al.*, 2016]. Given the network, we $i)$ insert 10% additional edges randomly to get G_1, and $ii)$ remove 15% of its original edges to get G_2.

Phone-Email contains two communication networks corresponding to phone calls and emails, respectively [Zhang *et al.*, 2020b]. G_1 contains 41,191 phone calls (i.e., edges) among 1,000 users (i.e., nodes), and similarly, G_2 contains 4,627 email communications among 1,003 users. There are 1,000 users appearing in both networks, and the ground-truth alignment matrix is provided.

It is easy to find that these datasets have different noise levels. The networks in the synthetic data are isomorphic and thus do not have any noise. The Arenas is a little noisy, in which G_1 and G_2 may have about 5% inconsistent edges. The Cora has medium noise, and the percentage of inconsistent edges increases to about 25%. The Phone-Email dataset is

Network Type	Method	Preserve Privacy	SynData ($I = 10$)			SynData ($I = 50$)			SynData ($I = 100$)		
			NC@1	NC@3	NC@5	NC@1	NC@3	NC@5	NC@1	NC@3	NC@5
Unweighted Directed	GWL	No	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$98.0_{\pm 0.0}$	$99.0_{\pm 0.0}$	$99.0_{\pm 0.0}$
	REGAL	No	$90.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$90.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$98.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
	FGWA	Yes	$60.0_{\pm 12.7}$	$68.0_{\pm 7.5}$	$82.0_{\pm 4.0}$	$36.0_{\pm 6.1}$	$68.8_{\pm 4.5}$	$85.2_{\pm 2.0}$	$18.8_{\pm 5.2}$	$49.4_{\pm 4.1}$	$68.2_{\pm 1.7}$
	CPP-SAHP	Yes	$82.0_{\pm 13.3}$	$88.0_{\pm 9.8}$	$88.0_{\pm 9.8}$	$60.0_{\pm 3.8}$	$81.2_{\pm 3.7}$	$84.0_{\pm 5.4}$	$28.2_{\pm 4.9}$	$51.8_{\pm 5.2}$	$57.4_{\pm 5.2}$
	CPP-THP	Yes	$86.0_{\pm 13.6}$	$92.0_{\pm 9.8}$	$96.0_{\pm 4.9}$	$58.8_{\pm 2.4}$	$80.0_{\pm 4.2}$	$82.8_{\pm 4.8}$	$25.2_{\pm 3.2}$	$52.2_{\pm 4.3}$	$57.4_{\pm 3.4}$
Weighted Directed	GWL	No	$80.0_{\pm 0.0}$	$80.0_{\pm 0.0}$	$80.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
	FGWA	Yes	$52.0_{\pm 11.7}$	$68.0_{\pm 13.3}$	$76.0_{\pm 10.2}$	$30.0_{\pm 8.8}$	$67.6_{\pm 8.0}$	$81.6_{\pm 5.0}$	$25.0_{\pm 6.3}$	$51.8_{\pm 4.3}$	$67.6_{\pm 4.6}$
	CPP-SAHP	Yes	$90.0_{\pm 6.3}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$61.6_{\pm 5.7}$	$78.8_{\pm 8.9}$	$80.8_{\pm 9.5}$	$31.8_{\pm 8.0}$	$50.8_{\pm 5.0}$	$54.6_{\pm 6.1}$
	CPP-THP	Yes	$92.0_{\pm 7.5}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$58.0_{\pm 4.0}$	$76.4_{\pm 8.9}$	$80.4_{\pm 8.9}$	$31.2_{\pm 7.3}$	$50.2_{\pm 6.5}$	$54.0_{\pm 5.7}$

Table 2: Node correctness on synthetic networks (%).

the most challenging one — the edges of the two networks are significantly unbalanced.

We generate a few event sequences for each dataset for the source G_1 and target network G_2, respectively, and infer the alignment matrix based on the event sequences. In particular, given a network $G(\mathcal{V}, \boldsymbol{A})$, we assume that its event sequences are generated by a Hawkes process [Zhou *et al.*, 2013] driven by the adjacency matrix $\boldsymbol{A}$, i.e., HP$(\boldsymbol{\mu}, \boldsymbol{\Phi})$, where $\boldsymbol{\mu} := \frac{1}{\|\boldsymbol{A}\mathbf{1}\|_\infty}\boldsymbol{A}\mathbf{1}$ and $\boldsymbol{\Phi} := \frac{0.8}{\|\boldsymbol{A}\|_2}\boldsymbol{A}\exp(t)$. Given the Hawkes process model, we simulate each event sequence independently in the time window $[0, 50]$ by Ogata's thinning algorithm [Ogata, 1981]. Table 1 shows the basic statistics of the datasets and the number of event sequences per network.

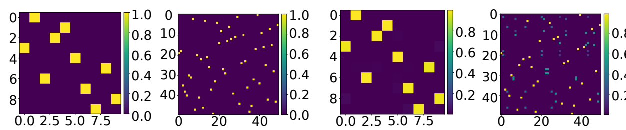
In our experiments, we compare our CPP-based alignment method with the following three competitors:

GWL [Xu *et al.*, 2019] and **REGAL** [Heimann *et al.*, 2018] are two state-of-the-art network alignment methods. GWL achieves an optimal transport-based graph matching algorithm that aligns networks and learns node embeddings jointly. REGAL learns the structure similarity between the networks by cross-network matrix factorization.

FGWA [Luo *et al.*, 2019] is a sequence-driven network alignment method. It jointly learns two Hawkes processes in a regularized maximum likelihood estimation framework. By penalizing the GW-based alignment loss [Titouan *et al.*, 2019] between the model parameters of the two Hawkes processes, this method learns an optimal transport matrix to infer the correspondence between the two networks.

Remark. On the one hand, GWL and REGAL work as "**oracles**" because of using topological information directly. On the other hand, FGWA applies the naïve strategy in (4) by setting the regularizer as the GW-based alignment loss, which works as the baseline of our method. We hope our CPP-based method can outperform the baseline and approach the oracles' performance. We evaluate the performance of the methods by the commonly-used top-K node correctness (denoted as NC@K) [Heimann *et al.*, 2018; Luo *et al.*, 2019; Xu *et al.*, 2019], which records the percentage of the nodes in G_2 whose correspondence nodes in G_1 are in the top-K lists inferred by an alignment method. In each experiment, we run all the methods in five trials, with different random seeds, and record the mean and standard deviation of NC@K.

We implement the methods in PyTorch and conduct experiments on NVIDIA GeForce RTX 3090. When implementing our method, we use SAHP [Zhang *et al.*, 2020a] and THP [Zuo *et al.*, 2020] as the backbone of our CPP model, respectively. Each backbone has four 4-head attention layers. The dimension of node embedding is 64, and the hidden dimension of the position-wise feed-forward network is 128. The coupled node embedding layer applies a parametric cost matrix by default. The weight of the IOT regularizer is set to be 100. For different datasets, we adjust the learning rate from 10^{-5} to 10^{-3} and the ϵ in the Sinkhorn matching module from 10^{-4} to 10^{-1}. For fairness, the hyperparameters of GWL, REGAL, and FGWA are optimized by grid search.



(a) Ground truth $\boldsymbol{P}$ (b) Approximated $\boldsymbol{P}_\gamma$

Figure 3: Illustrations of the ground truth and the approximated alignment matrices for the synthetic datasets.

5.2 Comparison Experiments

We first demonstrate the feasibility of our method on synthetic data. The alignment results of different methods on the synthetic data are shown in Table 2. Both REGAL and GWL outperform all the sequence-based alignment methods (including FGWA and ours). The gap is caused by i) the randomness introduced by stochastic event sequences; ii) the misspecification caused by the model assumption. However, as representative traditional alignment methods, REGAL and GWL are based on the network topology and thus cannot achieve privacy preservation. On the contrary, FGWA does not require the network topology directly, but it is only applicable for the Hawkes processes defined on networks and suffers from a high risk of model misspecification. As a result, the performance of FGWA is unsatisfactory compared to the oracles. Our CPP-based method outperforms FGWA consistently because the CPP model is more flexible and has better capacity. Additionally, our method is robust to the selection of backbone model — both CPP-SAHP and CPP-THP achieve encouraging alignment results. Besides the numerical results in Table 2, we further compare our approximated alignment matrices with the ground truth in Figure 3. In summary, our CPP-based method works better than FGWA, re-

Method	Preserve Privacy	Arenas NC@1	Arenas NC@5	Arenas NC@30	Coras NC@1	Coras NC@5	Coras NC@30	Phone-Email NC@1	Phone-Email NC@5	Phone-Email NC@30
GWL	No	$18.22_{\pm 0.34}$	$19.45_{\pm 0.15}$	$21.46_{\pm 0.17}$	$0.04_{\pm 0.00}$	$0.18_{\pm 0.01}$	$1.14_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.54_{\pm 0.17}$	$3.30_{\pm 0.26}$
REGAL	No	$46.56_{\pm 0.50}$	$66.94_{\pm 0.87}$	$84.83_{\pm 0.40}$	$7.99_{\pm 0.19}$	$16.71_{\pm 0.46}$	$31.14_{\pm 0.29}$	$0.10_{\pm 0.00}$	$0.54_{\pm 0.05}$	$2.90_{\pm 0.17}$
FGWA	Yes	$1.19_{\pm 0.24}$	$6.17_{\pm 0.33}$	$28.62_{\pm 0.63}$	$0.04_{\pm 0.00}$	$0.18_{\pm 0.00}$	$1.11_{\pm 0.00}$	$0.18_{\pm 0.10}$	$0.72_{\pm 0.35}$	$3.40_{\pm 0.79}$
CPP-SAHP	Yes	$1.96_{\pm 0.72}$	$7.31_{\pm 0.57}$	$27.79_{\pm 0.62}$	$0.55_{\pm 0.06}$	$1.64_{\pm 0.15}$	$5.52_{\pm 0.35}$	$0.12_{\pm 0.07}$	$0.86_{\pm 0.21}$	$4.62_{\pm 0.44}$
CPP-THP	Yes	$2.04_{\pm 0.47}$	$7.38_{\pm 0.40}$	$27.89_{\pm 0.77}$	$0.58_{\pm 0.77}$	$1.61_{\pm 0.12}$	$5.40_{\pm 0.37}$	$0.12_{\pm 0.12}$	$0.78_{\pm 0.26}$	$4.58_{\pm 0.26}$

Table 3: Node correctness on real-world datasets (%).

Cost matrix	IOT	Arenas NC@1	Arenas NC@5	Arenas NC@30
Non-parametric	$\tau = 0$	$0.07_{\pm 0.04}$	$0.42_{\pm 0.19}$	$2.15_{\pm 0.39}$
	$\tau = 0.01$	$1.78_{\pm 0.53}$	$6.56_{\pm 0.64}$	$12.93_{\pm 0.91}$
	$\tau = 1$	$1.76_{\pm 0.65}$	$7.12_{\pm 0.58}$	$24.19_{\pm 1.17}$
	$\tau = 100$	$1.89_{\pm 0.71}$	$7.24_{\pm 0.48}$	$25.48_{\pm 0.79}$
Parametric	$\tau = 0$	$0.02_{\pm 0.04}$	$0.39_{\pm 0.11}$	$2.45_{\pm 0.26}$
	$\tau = 0.01$	$1.94_{\pm 0.55}$	$7.59_{\pm 0.45}$	$27.51_{\pm 0.27}$
	$\tau = 1$	$1.87_{\pm 0.48}$	$7.31_{\pm 0.53}$	$27.59_{\pm 1.05}$
	$\tau = 100$	$\mathbf{2.04_{\pm 0.47}}$	$\mathbf{7.38_{\pm 0.40}}$	$\mathbf{27.89_{\pm 0.77}}$

Table 4: Impacts of various settings on node correctness (%)

SynData	#Sequences	NC@1	NC@3	NC@5
#Events	500	$7.00_{\pm 0.63}$	$13.40_{\pm 1.96}$	$19.00_{\pm 2.83}$
per	1,000	$7.40_{\pm 3.61}$	$16.20_{\pm 7.30}$	$21.20_{\pm 9.74}$
sequence	1,500	$11.20_{\pm 1.72}$	$22.60_{\pm 2.87}$	$31.00_{\pm 4.60}$
$\sim$266	2,000	$\mathbf{12.40_{\pm 0.80}}$	$\mathbf{25.00_{\pm 3.03}}$	$\mathbf{31.80_{\pm 2.99}}$
Arenas	**#Sequences**	**NC@1**	**NC@5**	**NC@30**
#Events	3,000	$1.59_{\pm 0.24}$	$6.80_{\pm 0.52}$	$24.30_{\pm 0.42}$
per	5,000	$1.80_{\pm 0.16}$	$6.98_{\pm 0.27}$	$25.87_{\pm 0.91}$
sequence	8,000	$1.67_{\pm 0.27}$	$6.66_{\pm 1.28}$	$25.71_{\pm 1.95}$
$\sim$159	10,000	$\mathbf{2.04_{\pm 0.47}}$	$\mathbf{7.38_{\pm 0.40}}$	$\mathbf{27.89_{\pm 0.77}}$

Table 5: The impacts of data sufficiency on node correctness (%)

ducing the gap to the oracles significantly. *These results mean that in the privacy-preserving network alignment problem, our method achieves a trade-off between alignment accuracy and privacy preservation.*

We further verify our above claim in real-world experiments. Table 3 records the alignment results of various methods on real-world datasets. In these experiments, all the methods suffer from performance degradation because the real-world networks are not isomorphic and often have noise, which makes the alignment tasks challenging. In the cases with light or medium noise (e.g., Arenas and Cora), the oracles still outperform the sequence-based alignment methods. However, in the case of heavy noise (e.g., Phone-Email), our CPP methods achieve the best performance. A potential reason for this interesting phenomenon is that the network topology becomes unreliable in highly noisy cases. Accordingly, the uncertainty caused by the event sequences is not dominant, which does not influence the results much.

5.3 Analytic Experiments

Take our CPP-THP as an example, we further analyze the rationality of our alignment method in the following analytic experiments. In Table 4, we show the alignment results of our CPP-THP method achieved under different settings, including the modeling strategy of the cost matrix and the utilization of the IOT regularizer. We can find that our method achieves comparable alignment results under different cost models, which further verifies its robustness to modeling strategy. On the contrary, the IOT regularizer impacts our method a lot. Table 4 shows that applying the IOT regularizer (i.e., $\tau \neq 0$) leads to significant improvements in the alignment results, which demonstrates the necessity of the regularizer and according to the rationality of our learning algorithm. Additionally, our method is robust to the weight of the IOT regularizer. As shown in Table 4, our alignment results are stable when τ changes in a wide range. In Table 5, we show the impacts of the number of sequences on our alignment results of SynData ($I = 100$) and Arenas, respectively. Increasing the number of training sequences helps to improve the alignment results consistently. The more data we have, the better alignment results we can obtain.

6 Conclusion and Future Work

We have proposed a CPP model and its learning algorithm, which aligns networks based on their event sequences rather than their topologies. The proposed method shows the feasibility of privacy-preserving network alignment, which achieves a trade-off between alignment accuracy and privacy preservation. In the future, we plan to further improve the scalability of the proposed method for large-scale applications. Additionally, we would like to make collaborations with network service providers and financial institutes and test our method in real-world scenarios.

Acknowledgments

Dr. Dixin Luo was supported by the National Natural Science Foundation of China (62102031) and the Young Scholar Program (XSQD-202107001) from Beijing Institute of Technology. Dr. Hongteng Xu was supported by the National Natural Science Foundation of China (62106271, 92270110), CAAI-Huawei MindSpore Open Fund, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China. He thanks the support from the Beijing Key Laboratory of Big Data Management and Analysis Methods, the Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative.

Contribution Statement

The first two authors of this work have equal contributions. Hongteng Xu is the correspondence author.

References

[Adams and Zemel, 2011] Ryan Prescott Adams and Richard S Zemel. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, 2011.

[Bacry *et al.*, 2015] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

[Bayati *et al.*, 2009] Mohsen Bayati, Margot Gerritsen, David F Gleich, Amin Saberi, and Ying Wang. Algorithms for large, sparse network alignment problems. In *ICDM*, 2009.

[Cuturi, 2013] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[Daley and Vere-Jones, 2007] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.

[Du *et al.*, 2016] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1555–1564, 2016.

[Fey *et al.*, 2020] Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. Deep graph matching consensus. *arXiv preprint arXiv:2001.09621*, 2020.

[Gilbert, 1959] Edgar N Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[Guo *et al.*, 2018] Ruocheng Guo, Jundong Li, and Huan Liu. Initiator: Noise-contrastive estimation for marked temporal point process. In *IJCAI*, pages 2191–2197, 2018.

[Hashemifar *et al.*, 2016] Somaye Hashemifar, Qixing Huang, and Jinbo Xu. Joint alignment of multiple protein–protein interaction networks via convex optimization. *Journal of Computational Biology*, 23(11):903–911, 2016.

[Hawkes, 1971] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[Heimann *et al.*, 2018] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 117–126, 2018.

[Huang *et al.*, 2016] Jiaxiang Huang, Maoguo Gong, and Lijia Ma. A global network alignment method using discrete particle swarm optimization. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(3):705–718, 2016.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Koutra *et al.*, 2013] Danai Koutra, Hanghang Tong, and David Lubensky. Big-align: Fast bipartite graph alignment. In *2013 IEEE 13th international conference on data mining*, pages 389–398. IEEE, 2013.

[Leskovec and Sosič, 2016] Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Trans. Intell. Syst. Technol.*, 8(1), jul 2016.

[Li *et al.*, 2019] Ruilin Li, Xiaojing Ye, Haomin Zhou, and Hongyuan Zha. Learning to match via inverse optimal transport. *Journal of machine learning research*, 20, 2019.

[Liu *et al.*, 2016] Li Liu, William K Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In *IJCAI*, volume 16, pages 1774–80, 2016.

[Luo *et al.*, 2019] Dixin Luo, Hongteng Xu, and Lawrence Carin. Fused gromov-wasserstein alignment for hawkes processes. *arXiv preprint arXiv:1910.02096*, 2019.

[Malod-Dognin and Pržulj, 2015] Noël Malod-Dognin and Nataša Pržulj. L-GRAAL: Lagrangian graphlet-based network aligner. *Bioinformatics*, 31(13):2182–2189, 2015.

[Mei and Eisner, 2017] Hongyuan Mei and Jason M Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In *NIPS*, 2017.

[Mei *et al.*, 2020] Hongyuan Mei, Tom Wan, and Jason Eisner. Noise-contrastive estimation for multivariate point processes. *Advances in neural information processing systems*, 33:5204–5214, 2020.

[Mohammadi *et al.*, 2017] Shahin Mohammadi, David F Gleich, Tamara G Kolda, and Ananth Grama. Triangular alignment TAME: A tensor-based approach for higher-order network alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 14(6):1446–1458, 2017.

[Mohler *et al.*, 2011] George O Mohler, Martin B Short, P Jeffrey Brantingham, Frederic Paik Schoenberg, and George E Tita. Self-exciting point process modeling of crime. *Journal of the american statistical association*, 106(493):100–108, 2011.

[Nassar *et al.*, 2018] Huda Nassar, Nate Veldt, Shahin Mohammadi, Ananth Grama, and David F Gleich. Low rank spectral network alignment. In *WWW*, 2018.

[Neyshabur *et al.*, 2013] Behnam Neyshabur, Ahmadreza Khadem, Somaye Hashemifar, and Seyed Shahriar Arab. NETAL: A new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, 29(13):1654–1662, 2013.

[Ogata, 1981] Yosihiko Ogata. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, 1981.

[Patro and Kingsford, 2012] Rob Patro and Carl Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28(23):3105–3114, 2012.

[Pourhabibi *et al.*, 2020] Tahereh Pourhabibi, Kok-Leong Ong, Booi H Kam, and Yee Ling Boo. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133:113303, 2020.

[Sinkhorn and Knopp, 1967] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[Sun *et al.*, 2015] Yihan Sun, Joseph Crawford, Jie Tang, and Tijana Milenković. Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In *International Workshop on Algorithms in Bioinformatics*, pages 16–39, 2015.

[Sun *et al.*, 2022] Ying Sun, Wenjun Wang, Nannan Wu, Chaochao Liu, Siddharth Bhatia, Yang Yu, and Wei Yu. Aaan: Anomaly alignment in attributed networks. *Knowledge-Based Systems*, 249:108944, 2022.

[Titouan *et al.*, 2019] Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pages 6275–6284. PMLR, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Vijayan *et al.*, 2015] Vipin Vijayan, Vikram Saraph, and T Milenković. MAGNA++: Maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics*, 31(14):2409–2411, 2015.

[Wang *et al.*, 2019a] Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. Hierarchical matching network for crime classification. In *proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 325–334, 2019.

[Wang *et al.*, 2019b] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3056–3065, 2019.

[Xu *et al.*, 2016a] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning Granger causality for Hawkes processes. In *ICML*, 2016.

[Xu *et al.*, 2016b] Hongteng Xu, Weichang Wu, Shamim Nemati, and Hongyuan Zha. Patient flow prediction via discriminative learning of mutually-correcting processes. *IEEE transactions on Knowledge and Data Engineering*, 29(1):157–171, 2016.

[Xu *et al.*, 2019] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, pages 6932–6941. PMLR, 2019.

[Yan *et al.*, 2021] Yuchen Yan, Si Zhang, and Hanghang Tong. Bright: A bridging algorithm for network alignment. In *Proceedings of the Web Conference 2021*, pages 3907–3917, 2021.

[Yang *et al.*, 2016] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[Zanfir and Sminchisescu, 2018] Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *CVPR*, 2018.

[Zhang and Tong, 2016] Si Zhang and Hanghang Tong. Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1345–1354, 2016.

[Zhang *et al.*, 2019] Si Zhang, Hanghang Tong, Ross Maciejewski, and Tina Eliassi-Rad. Multilevel network alignment. In *The World Wide Web Conference*, pages 2344–2354, 2019.

[Zhang *et al.*, 2020a] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR, 2020.

[Zhang *et al.*, 2020b] Si Zhang, Hanghang Tong, Jie Tang, Jiejun Xu, and Wei Fan. Incomplete network alignment: Problem definitions and fast solutions. *ACM Transactions on Knowledge Discovery from Data*, 14(4):1–26, 2020.

[Zhang *et al.*, 2020c] Wei Zhang, Thomas Panum, Somesh Jha, Prasad Chalasani, and David Page. Cause: Learning granger causality from event sequences using attribution methods. In *International Conference on Machine Learning*, pages 11235–11245. PMLR, 2020.

[Zhang *et al.*, 2021] Si Zhang, Hanghang Tong, Long Jin, Yinglong Xia, and Yunsong Guo. Balancing consistency and disparity in network alignment. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2212–2222, 2021.

[Zhou *et al.*, 2013] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes. In *AISTATS*, 2013.

[Zhu and Xie, 2022] Shixiang Zhu and Yao Xie. Spatiotemporal-textual point processes for crime linkage detection. *The Annals of Applied Statistics*, 16(2):1151–1170, 2022.

[Zuo *et al.*, 2020] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.