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

Dixin Luo\textsuperscript{1*}, Haoran Cheng\textsuperscript{1*}, Qingbin Li\textsuperscript{1}, Hongteng Xu\textsuperscript{2,3}

\textsuperscript{1}School of Computer Science and Technology, Beijing Institute of Technology
\textsuperscript{2}Gaoling School of Artificial Intelligence, Renmin University of China
\textsuperscript{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-
our method is compatible with various point process back-bones [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.

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],

---

1For 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].
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 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., \( \{G_k(\mathcal{V}_k, A_k)\}_{k=1}^2 \). For the \( k \)-th network, \( \mathcal{V}_k = \{v_k^i\}_{i=1}^{N_k} \) is the node set, and \( A_k \in \mathbb{R}^{T_k \times T_k} \) 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 \( A_k \)'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 \( S_k = \{t_{k,m}^i, v_{k,m}^i\} \subseteq [0, T] \times \mathcal{V}_k \) for \( k = 1, 2 \). Here, \( S_k \) represents the event sequence generated by the \( k \)-th network, and \( \{t_{k,m}^i, v_{k,m}^i\} \) represents the \( m \)-th event of the node \( v_{k,m}^i \in \mathcal{V}_k \) at the timestamp \( t_{k,m}^i \in [0, T] \), 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.,

\[
S_k \sim N_k(t, A_k), \quad \text{for} \quad k = 1, 2, \quad (1)
\]

where \( N_k(t, A_k) = \{N_k(t, A_k)\}_{v \in \mathcal{V}_k} \) is the counting process corresponding to the temporal point process, whose parameters are determined by the adjacency matrix \( A_k \). \( N_k(t, A_k) \) 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.,

\[
\lambda_k^v(t) dt = \mathbb{E}[dN_k^v(t) | H_k^t], \quad \text{for} \quad v \in \mathcal{V}_k, \quad (2)
\]

where \( v \) indicates the node index belonging to a node set \( \mathcal{V}_k \), and \( H_k^t = \{t_{k,m}^i, v_{k,m}^i\} \in S_k | t_{k,m}^i < t \) represents the historical events till time \( t \).

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

\[
A_2 = PA_1P^T + E, \quad (3)
\]

where the alignment matrix \( P \in \{0, 1\}^{N_2 \times N_1} \) indicates the correspondence between the two networks’ nodes, and \( E \in \mathbb{R}^{N_2 \times N_1} \) represents the unknown random noise. Given \( \{S_k\}_{k=1}^2 \), we would like to infer \( P \) 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 naive way is solving the following learning task:

\[
\min_{\{A_k\}_{k=1}^2} \mathbb{E}_{(S_k, A_k)} \sum_{k=1}^2 \mathcal{L}(S_k, A_k) + \tau R(A_1, A_2, P), \quad (4)
\]

where \( \mathcal{L}(S_k, A_k) \) is the negative log-likelihood (NLL) of the sequence \( S_k \), which is defined as

\[
\sum_v \int_0^T \lambda_k^v(s) ds - \sum_{m=1}^{M_k} \log \lambda_k^{v_m}(t_{k,m}^m), \quad (5)
\]

and \( R(A_1, A_2, P) \) 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 \( \|A_2 - PA_1P^T\|_2^2 \) 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.,

\[
R(A_1, A_2, P) = \max_{\Omega \in \Omega} \mathbb{F}_{\Omega}(A_2, PA_1, P), \quad \text{where} \quad \Omega \text{ is a doubly-stochastic constraint on } P \text{ and } \langle \cdot, \cdot \rangle \text{ indicates the inner product of matrices.}
\]

Unfortunately, this naive 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 dynamics jointly based on the same conditional intensity function instead of modeling them independently. Specifically, given a timestamp \( t \) and the historical events \( H_t \) (from either \( S_1 \) or \( S_2 \)), we have

\[
\lambda_v(t) = g(t, H_t; \{v_k^i\}_{k=1}^2, \theta), \quad \text{for} \quad v \in \mathcal{V}_1 \cup \mathcal{V}_2 \quad (6)
\]

where \( V_k = [v_k^i] \in \mathbb{R}^{D_k \times T_k} \) 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 \( \{v_k^i\}_{k=1}^2 \) that are specified for each network, and ii) the parameter \( \theta \) 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 $H_P(\mu, \Phi)$, its intensity function $\lambda_v(t) = \mu_v + \sum_{i, j \neq v} \phi_{v, i}(t - t_i)$ consists of a base intensity $\mu_v$ and a set of impact functions $\Phi = [\phi_{v, i}(t)]$. Each impact function is often modeled based on the adjacency matrix and a decay function, i.e., $\phi_{v, i}(t) = a_{v, i} e^{-\lambda t}$, where $a_{v, i}$ is the element of $A$. Our CPP parameterizes $A$ as a bi-linear model $V^T W V$ and $\mu$ as a linear model $V^T w$, and accordingly, $\theta = \{W \in \mathbb{R}^{D \times D}, w \in \mathbb{R}^D\}$.

**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 RMTTPP [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., $V_2 = V_1 P^T$. 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

$$V_2 := V_1 P^T,$$

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

$$P_\gamma = I_2 \text{Sinkhorn}(C_\gamma, \epsilon) = I_2 \arg \min_{T \in \Pi(\frac{1}{T_2} \mathbf{1}_I, \frac{1}{T_1} \mathbf{1}_I)} \langle C_\gamma, T \rangle + \epsilon \langle T, \log T \rangle,$$

where Sinkhorn$(C_\gamma, \epsilon)$ denotes the matching matrix 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, $C_\gamma \in \mathbb{R}^{I_2 \times I_1}$ is the grounding cost matrix of the EOT problem, which is with parameter $\gamma$. In this study, we consider two implementations of $C_\gamma$: 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.

The optimal solution of the EOT problem, denoted as $T^*$ is a doubly-stochastic matrix, i.e., $T^* \in \Pi(\frac{1}{T_2} \mathbf{1}_I, \frac{1}{T_1} \mathbf{1}_I) = \{T \in \mathbb{R}^{I_2 \times I_1} | T \mathbf{1}_{I_1} = \frac{1}{T_2} \mathbf{1}_I, T^T \mathbf{1}_{I_2} = \frac{1}{T_1} \mathbf{1}_I \}$. Therefore, $P_\gamma = I_2 T^*$ is a transition matrix, which provides a probabilistic approximation of the alignment matrix $P$. In particular, the element of $P_\gamma$, denoted as $p(v|v'; \gamma)$, represents the conditional probability of $v \in V_2$ given $v' \in V_1$. 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.,

$$\bar{\lambda}_v(T) = N_v(T)/T, \forall v \in V_1 \cup V_2.$$

(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 $\{S_k\}_{k=1}^2$.
2: **Hyperparameters:** The number of iterations $M$, the weight of the IOT regularizer $\tau$.
3: Compute the average intensity for each node in $V_1 \cup V_2$ and construct $C$.
4: Compute the priori alignment matrix $P_0$ via Algorithm 1 and fix it as a constant.
5: for $m = 1, ..., M$ do
6: Compute the loss function in (10) and update the parameters by Adam [Kingma and Ba, 2014].
7: end for
8: **Output:** Learned alignment matrix $P^\gamma$.

for the alignment matrix. In particular, we first calculate the average intensity for each node and construct a cost matrix $C = [c_{v,v'}] \in \mathbb{R}_+^{I_2 \times I_1}$, where $c_{v,v'} = [\bar{\lambda}_v(T) - \bar{\lambda}_{v'}(T)]$ for $v \in V_2$ and $v' \in V_1$. Then, we reuse the Sinkhorn matching module in Algorithm 1, computing a priori alignment matrix, denoted as $P_0 = I_2 \text{Sinkhorn}(C, \epsilon)$.

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., $\bar{\lambda}_v(T) \rightarrow \mathbb{E}[\lambda(T)]$ 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:

$$
\min_{V_1, P_1, \theta} L(S_1 \cup S_2; V_1, P_1, \theta) + \tau \text{KL}(P_1 \| P_0),
$$

(10)

where $S_1 \cup S_2$ is the collection of the event sequences of the two networks. They are modeled jointly by our CPP model, and $L$ is the negative log-likelihood defined in (5). $\text{KL}(P_1 \| P_0)$ penalizes the KL-divergence between the approximated alignment matrix and its prior. From the viewpoint of optimal transport, $\frac{1}{T_1} P_1$ is a transport matrix derived based on a learnable cost matrix $C_\gamma$, while $\frac{1}{T_2} P_0$ 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.

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 Erdős-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 $P$ 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 Sosić, 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 1,003 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
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 $G(V, A)$, we assume that its event sequences are generated by a Hawkes process [Zhou et al., 2013] driven by the adjacency matrix $A$, i.e., $HP(\mu, \Phi)$, where $\mu := \frac{1}{|A|} A_1$ and $\Phi := \frac{1}{|A|} A \exp(t)$. 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 naive 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-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 100. For different datasets, we adjust the learning rate from $10^{-5}$ to $10^{-3}$ and the $\epsilon$ 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.

### 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-

![Figure 3: Illustrations of the ground truth and the approximated alignment matrices for the synthetic datasets.](image-url)
Table 3: Node correctness on real-world datasets (%).

<table>
<thead>
<tr>
<th>Method</th>
<th>Preserve</th>
<th>Cost matrix</th>
<th>Arenas</th>
<th>Coras</th>
<th>Phone-Email</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IOT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ = 0</td>
<td>0.01</td>
<td>1.00</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ = 0.01</td>
<td>1.78 ± 1.75</td>
<td>6.56 ± 0.94</td>
<td>12.93 ± 0.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ = 1</td>
<td>1.76 ± 0.65</td>
<td>7.12 ± 0.58</td>
<td>24.19 ± 1.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ = 100</td>
<td>1.89 ± 0.71</td>
<td>7.24 ± 0.48</td>
<td>25.48 ± 0.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ = 1000</td>
<td>2.04 ± 0.47</td>
<td>7.38 ± 0.40</td>
<td>27.89 ± 0.77</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Preserve</th>
<th>#Events</th>
<th>Coras</th>
<th>Phone-Email</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>7.00 ± 1.36</td>
<td>13.40 ± 1.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,000</td>
<td>7.40 ± 3.61</td>
<td>16.20 ± 7.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,500</td>
<td>11.20 ± 7.12</td>
<td>22.60 ± 2.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>~266</td>
<td>12.40 ± 0.80</td>
<td>25.00 ± 3.08</td>
</tr>
</tbody>
</table>

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., τ ≠ 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 yields 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


