Preliminaries

Hawkes Process

The Hawkes process (Hawkes 1971) is a significant method of temporal point process (Xia, Li, and Li 2023; Shen et al. 2014; Tsuchida, Ong, and Sejdinovic 2024). Its core mechanism posits that historical events exert a cumulative influence on the occurrence of current events. Specifically, recent historical events have a greater impact on the probability of a current event occurring, whereas more distant events have a lesser effect. At time t, this cumulative influence is captured through a conditional intensity function, defined as follows:

$$\lambda(t) = \mu + \int_{-\infty}^{t} \kappa(t - t')dN_{t'}(\cdot) \tag{12}$$

where μ is the base intensity function without any external influences, and $\kappa(\cdot)$ is a kernel function used to quantify the time decay effect of historical events on current events.

Variational Autoencoder

The central tenet of variational autoencoder methods is the iterative optimization of the ELBO to ensure that the introduced posterior distribution $q(\cdot)$ approximates the true yet unknown prior distribution $p(\cdot)$. Given a node u randomly sampled from the sample space, a common approach is to model its representation z as following a Gaussian distribution. The posterior distribution is incrementally refined to better approximate the prior distribution by maximizing the ELBO, as follows:

$$\begin{aligned}\log p(x) &\geq \mathbf{E}_{z\sim q(z|x)} \log \frac{p(x \mid z)p(z)}{q(z \mid x)} \\ &= \underbrace{\int_z q(z \mid x) log p(x \mid z)dz}_{Generation} - \underbrace{KL(q(z \mid x)\|p(z))}_{Infernece}\end{aligned} \tag{13}$$

where x denotes the input data of node u, and KL stands for Kullback-Leibler Divergence.

Datasets

We used six datasets for our experimental protocol: DBLP, Brain, Patent, School, arXivAI and arXivCS. We provide the statistics of these datasets as follows, where node homogeneity ratios $\mathcal{H}_n$ and edge homogeneity ratios $\mathcal{H}_e$ are calculated by Eq. (14) and Eq. (15), respectively.

Table 3: Statistics of the used datasets.

Datasets	$\mathcal{V}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{C}$	$\mathcal{H}_n$	$\mathcal{H}_e$
DBLP	28,085	236,894	162,441	10	0.6724	0.6201
Brain	5,000	1,955,488	1,751,910	10	0.2202	0.2200
Patent	12,214	41,916	41,915	6	0.6538	0.6704
School	327	188,508	5,802	9	0.7215	0.6935
arXivAI	69,854	699,206	699,198	5	0.7453	0.7293
arXivCS	169,343	1,166,243	1,166,237	40	0.6352	0.6542

$$\mathcal{H}_n = \frac{1}{|\mathcal{V}|} \sum_{u\in\mathcal{V}} \frac{|\{v \in N(u) : \mathcal{C}(u) = \mathcal{C}(v)\}}{|N(u)|} \tag{14}$$

$$\mathcal{H}_e = \frac{|(u, v) \in \mathcal{E}, \mathcal{C}(u) = \mathcal{C}(v)|}{|\mathcal{E}|} \tag{15}$$

Complexity Analysis

Since we treat the continuous dynamic network as an event stream, it can be trained in batches. In this scenario, the training time overhead is related to the number of interactions between nodes, but not the number of nodes. Consequently, the time complexity of our method is $O(\mathcal{E})$, rather than $O(|\mathcal{V}^2|)$. As noted by (Liu et al. 2024b), in most cases $O(\mathcal{E})$ is significantly less than $O(|\mathcal{V}^2|)$, and the two complexities are equal only if the network is a fully connected network with a single interaction between nodes. The case where $O(\mathcal{E}) > O(|\mathcal{V}^2|)$ occurs only in a few specific scenarios, such as a small network subjected to malicious information bombardment.

It is worth mentioning that during training, the node information for each batch can be quickly retrieved directly through the first-order index. In recent years, (Wen and Fang 2022) proposed a Hawkes process-based GCN strategy, as follows:

$$\begin{aligned}\lambda_{u,v}(t) &= f(\mathbf{Z}_u^{t,l}, \mathbf{Z}_v^{t,l}) \\ &= f(\phi(\mathbf{Z}_u^{t,l-1}\mathcal{W}_{self}^{l} + \sum_{v,t'\in N_t(u)} \mathbf{Z}_v^{t',l-1}\mathcal{W}_{hist}^{l}\kappa(t - t')), \\ &\quad \phi(\mathbf{Z}_v^{t,l-1}\mathcal{W}_{self}^{l} + \sum_{u,t'\in N_t(v)} \mathbf{Z}_u^{t',l-1}\mathcal{W}_{hist}^{l}\kappa(t - t')))\end{aligned} \tag{16}$$

where f is transfer function, and l is the convolutional layer. Obviously, by slightly modifying Eq. (16), one can calculate the mean and variance of the Gaussian distribution. This modification allows node representations to be obtained based on the reparameterization technique. Although this approach may be effective, it requires significantly more time for node information sampling in each batch compared to our method. Specifically, due to the non-adjacent structure of continuous dynamic networks, computing the second-order convolution necessitates the use of multi-hop indexes. For example, each batch must additionally include two-hop neighbor information about target nodes. This requirement makes the node information sampling process cumbersome and cannot be mitigated by pre-training. Figure 4 illustrates the time overhead of sampling node information in one epoch based on the vanilla Hawkes process and on Eq. (16). Considering the factor, we do not focus on Eq. (16) in our inference process.

Proof Equations in the Generation Component Converge to Approximate Numerical Solutions

Proof. The generation components in Eq. (5) and Eq. (13) converge to the approximate numerical solution. First, the generation component in Eq. (13) can be expanded using cross-entropy. To intuitively describe the proof process, we omit certain coefficients and summation calculations involved in the cross-entropy, as follows:

$$\log(\phi(\mathbf{Z}_u, \mathbf{Z}_v)) + \log(1 - \phi(\mathbf{Z}_u, \mathbf{Z}_v)) \tag{17}$$

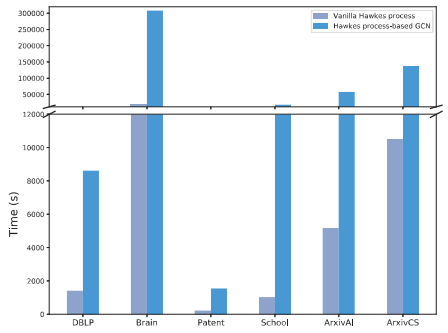

Figure 4: Time cost statistics.

where $\phi(x) = \frac{1}{1+e^{-x}}$ represents the sigmoid function. The term $(\cdot, \cdot)$ is typically calculated using either the dot product (Mrabah, Bouguessa, and Ksantini 2022) or other similarity measures (Hou et al. 2022). In this context, we assume it is computed using cosine similarity. It is worth noting that the dot product is also applicable for this proof. Subsequently, we compute the gradient of the first term in this equation as follows:

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{Z}_u} \log(\phi(\mathbf{Z}_u, \mathbf{Z}_v)) \\
&= \frac{\partial}{\partial \mathbf{Z}_u} \log(\frac{1}{1+e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)}}) \\
&= \frac{\partial}{\partial \mathbf{Z}_u} (-\log(1+e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)})) \\
&= \frac{1}{1+e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)}} \frac{\partial}{\partial \mathbf{Z}_u} \cos(\mathbf{Z}_u, \mathbf{Z}_v)
\end{aligned} \tag{18}$$

Similarly, we calculate the gradient of the second term as follows:

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{Z}_u} \log(1-\phi(\mathbf{Z}_u, \mathbf{Z}_v)) \\
&= \frac{\partial}{\partial \mathbf{Z}_u} \log(1-\frac{1}{1+e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)}}) \\
&= \frac{\partial}{\partial \mathbf{Z}_u} \log(\frac{e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)}}{1+e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)}}) \\
&= \frac{\partial}{\partial \mathbf{Z}_u} (-\cos(\mathbf{Z}_u, \mathbf{Z}_v) - \log(1+e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)})) \\
&= -\frac{\partial}{\partial \mathbf{Z}_u} \cos(\mathbf{Z}_u, \mathbf{Z}_v) - \frac{e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)}}{1+e^{-\cos(\mathbf{Z}_u, \mathbf{Z}_v)}} \frac{\partial}{\partial \mathbf{Z}_u} (-\cos(\mathbf{Z}_u, \mathbf{Z}_v)) \\
&= -\frac{\partial}{\partial \mathbf{Z}_u} \cos(\mathbf{Z}_u, \mathbf{Z}_v) - (1-\phi(\cos(\mathbf{Z}_u, \mathbf{Z}_v))) \frac{\partial}{\partial \mathbf{Z}_u} (\cos(\mathbf{Z}_u, \mathbf{Z}_v))
\end{aligned} \tag{19}$$

Finally, we calculate the gradient in Eq. (5). Note that the gradients of the first and second terms in this equation are mathematically identical. Therefore, we only need to calculate the first one, as follows:

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{Z}_u} |\mathbf{X}_{u,v} - \cos(\mathbf{Z}_u, \mathbf{Z}_v)| \\
&= -\frac{1}{2} (\mathbf{X}_{u,v} - \cos(\mathbf{Z}_u, \mathbf{Z}_v))^{-1/2} \frac{\partial}{\partial \mathbf{Z}_u} \cos(\mathbf{Z}_u, \mathbf{Z}_v)
\end{aligned} \tag{20}$$

Similarly, we calculate the gradient of the third term as follows:

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{Z}_u} |0 - \cos(\mathbf{Z}_u, \mathbf{Z}'_v)| \\
&= \frac{1}{2} \cos(\mathbf{Z}_u, \mathbf{Z}'_v)^{-1/2} \frac{\partial}{\partial \mathbf{Z}_u} \cos(\mathbf{Z}_u, \mathbf{Z}'_v)
\end{aligned} \tag{21}$$

Clearly, the gradients of Eq. (17) and Eq. (5) both depend on the gradient of $\cos(\cdot, \cdot)$. This indicates that the gradient directions of these two optimization equations are consistent. Consequently, from a mathematical standpoint, Eq. (17) and Eq. (5) converge towards similar solutions during the optimization process.

This completes the proof. □