# A Theoretical Results

**Lemma A.1.** *Let $e_0 = 0 < e_1 < \cdots < e_{M-1} < T$ be a sequence following a Sequential Survival process for node pair $(i, j) \in \mathcal{V}^2$. Then, the average squared distance between nodes during interval $[e_m, e_{m+1})$ associated with survival function $S_m(\cdot)$ and state $s_m \in \{-1, 1\}$ can be bounded by*

$$b_m(-1) \leq \frac{1}{(e_{m+1} - e_m)} \int_{e_m}^{e_{m+1}} \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 dt \leq b_m(+1)$$

*where $b_m(s) := -2s \log(e_{m+1} - e_m) - s \log S(e_{m+1}) - s\beta(s)$.*

*Proof.* Let $e_0 = 0 < e_1 < \cdots < e_{M-1} < T$ be a sequence following a Sequential Survival process for node pair $(i, j) \in \mathcal{V}^2$, so we have an associated survival function, $S_m(t)$, for each $m$'th interval $[e_m, e_{m+1})$ with state $s_m \in \mathcal{S} := \{-1, 1\}$ due to the construction of the Sequential Survival process. Then, bounds for the survival function, $S_m(t)$, can be given by using Markov's inequality as:

$$\begin{aligned} S_m(t) = \mathbb{P}(T_m \geq e_{m+1}) &\leq \frac{\mathbb{E}[T_m]}{e_{m+1} - e_m} \\ &= \frac{1}{(e_{m+1} - e_m)} \frac{1}{\left( \int_{e_m}^{e_{m+1}} \exp\left( \beta(s) + s\|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 \right) \mathrm{dt} \right)} \end{aligned}$$

where $T_m$ is the random variable showing event time after time $e_m$. The last line implies that

$$\begin{aligned} \frac{1}{(e_{m+1} - e_m)\, S_m(t)} &\geq \int_{e_m}^{e_{m+1}} \exp\left( \beta(s_m) + s\|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 \right) dt \\ &= \exp(\beta(s_m)) \int_{e_m}^{e_{m+1}} \exp\left( s_m\|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 \right) dt \end{aligned} \tag{8}$$

Furthermore, we can apply Jensen's inequality for $\exp(s \cdot x)$ term since it is a convex function for any $s \in \{-1, 1\}$ value. Hence, we can write

$$\frac{1}{(e_{m+1} - e_m)} \int_{e_m}^{e_{m+1}} \exp\left( s_m\|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 \right) dt \geq \exp\left( \frac{s_m}{(e_{m+1} - e_m)} \int_{e_m}^{e_{m+1}} \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 dt \right) \tag{9}$$

By combining Eq. (8) and Eq. (9), we can obtain the following inequality:

$$\frac{1}{(e_{m+1} - e_m)^2 S_m(t)} \geq \exp(\beta(s_m)) \exp\left( \frac{s_m}{(e_{m+1} - e_m)} \int_{e_m}^{e_{m+1}} \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 dt \right).$$

The reorganization of the terms after taking the logarithm of the inequality yields:

$$2 \log(e_{m+1} - e_m) + \log S_m(e_{m+1}) + \beta(s_m) \leq -\frac{s_m}{(e_{m+1} - e_m)} \int_{e_m}^{e_{m+1}} \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 \mathrm{dt}$$

Finally, we can conclude that

$$b_m(-1) \leq \frac{1}{(e_{m+1} - e_m)} \int_{e_m}^{e_{m+1}} \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 dt \leq b_m(+1)$$

where $b_m(s) := -2s \log(e_{m+1} - e_m) - s \log S(e_{m+1}) - s\beta(s)$. $\square$

**Lemma A.2** (Integral Computation). *The integral of the hazard function for a given pair $(i, j) \in \mathcal{V}^2$ having constant velocities and states during the interval $(t_l, t_u)$ is equal to*

$$\int_{t_l}^{t_u} \lambda_{ij}(s,t) = \frac{\sqrt{\pi}}{2\|\Delta \mathbf{v}_{ij}\|} \exp\left(\beta(s) + s\|\Delta \mathbf{x}_{ij}\|^2 - s\rho_{ij}^2\right) \times E_{ij}(s, \tau(t_l), \tau(t_u))$$

*where $\rho_{ij} := \langle \Delta \mathbf{v}_{ij}, \Delta \mathbf{x}_{ij} \rangle / \|\Delta \mathbf{v}_{ij}\|$ and $E(\tau(t_l), \tau(t_u), s)$ is defined by*

$$E_{ij}(s, t_l, t_u) := \begin{cases} \mathrm{erf}\big(\tau(t_u)\big) - \mathrm{erf}\big(\tau(t_l)\big) & s = +1 \\ \mathrm{erfi}\big(\tau(t_u)\big) - \mathrm{erfi}\big(\tau(t_l)\big) & s = -1 \end{cases}$$

*for position difference, $\Delta \mathbf{x}_{ij} := \mathbf{r}_i(t_l) - \mathbf{r}_j(t_l)$, at time $t_l$ and velocity difference $\Delta \mathbf{v}_{ij} := \mathbf{v}_i - \mathbf{v}_j$. The function $\tau : (t_t, t_u) \to \mathbb{R}$ is defined as $\|\Delta \mathbf{v}_{ij}\| t + \rho_{ij}$, and $\mathrm{erf}(\cdot)$, and $\mathrm{erfi}(\cdot)$ represent the error and the imaginary error functions.*

*Proof.* Since it is supposed that the pair of nodes $(i, j) \in \mathcal{V}^2$ have constant velocities and states over the given interval $(t_l, t_u)$, the integral can be reexpressed in the following way:

$$\begin{aligned} \int_{t_l}^{t_u} \lambda_{ij}(s,t) dt &= \int_{t_l}^{t_u} \exp\Big(\beta(s) + s\|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2\Big) dt \\ &= \int_{t_l}^{t_u} \exp\Big(\beta(s) + s\|\Delta \mathbf{x}_{ij} + \Delta \mathbf{v}_{ij}(t - t_l)\|^2\Big) dt \\ &= \int_{0}^{t_u - t_l} \exp\Big(\beta(s) + s\|\Delta \mathbf{x}_{ij} + \Delta \mathbf{v}_{ij} t\|^2\Big) dt \end{aligned} \tag{10}$$

where $\Delta \mathbf{x}_{ij} := \mathbf{r}_i(t_l) - \mathbf{r}_j(t_l)$ and $\Delta \mathbf{v}_{ij} := \mathbf{v}_i - \mathbf{v}_j$ indicate the differences between the positions and velocities, and the last line, Eq. (10), is obtained by changing the boundaries of the integral. Since the bias term, $\beta(s)$, does not vary by time, it can be moved outside the integral term:

$$\int_{0}^{t_u - t_l} \exp\left(\beta(s) + s\|\Delta \mathbf{x}_{ij} + \Delta \mathbf{v}_{ij} t\|^2\right) dt = \exp\left(\beta(s)\right) \int_{0}^{t_u - t_l} \exp\Big(s\|\Delta \mathbf{x}_{ij} + \Delta \mathbf{v}_{ij} t\|^2\Big) dt \tag{11}$$

Then, we can rewrite the integral term as follows:

$$\begin{aligned} \int_{0}^{t_u - t_l} \exp\left(s\|\Delta \mathbf{x}_{ij} + \Delta \mathbf{v}_{ij} t\|^2\right) &= \int_{0}^{t_u - t_l} \exp\Big(s\|\Delta \mathbf{x}_{ij}\|^2 + 2s\big\langle \Delta \mathbf{x}_{ij}, \Delta \mathbf{v}_{ij} \big\rangle t + s\|\Delta \mathbf{v}_{ij}\|^2 t^2\Big) dt \\ &= \int_{0}^{t_u - t_l} \exp\Big(s\|\Delta \mathbf{x}_{ij}\|^2 - s\rho_{ij}^2 + s\left(\|\Delta \mathbf{v}_{ij}\| t + \rho_{ij}\right)^2\Big) dt \\ &= \exp\Big(s\|\Delta \mathbf{x}_{ij}\|^2 - s\rho_{ij}^2\Big) \int_{0}^{t_u - t_l} \exp\Big(s\big(\|\Delta \mathbf{v}_{ij}\| t + \rho_{ij}\big)^2\Big) dt \end{aligned} \tag{12}$$

where $\rho_{ij} := \langle \Delta \mathbf{v}_{ij}, \Delta \mathbf{x}_{ij} \rangle / \|\Delta \mathbf{v}_{ij}\|$. A substitution $y = \|\Delta \mathbf{v}_{ij}\| t + \rho_{ij}$ gives us $dy = \|\Delta \mathbf{v}_{ij}\| dt$, so we can write

$$\begin{aligned} \int_{0}^{t_u - t_l} \exp\left(s\left(\|\Delta \mathbf{v}_{ij}\| t + \rho_{ij}\right)^2\right) dt &= \frac{1}{\|\Delta \mathbf{v}_{ij}\|} \int_{\tau(t_l)}^{\tau(t_u)} \exp\left(sy^2\right) dy \\ &= \frac{1}{\|\Delta \mathbf{v}_{ij}\|} \frac{\sqrt{\pi}}{2} \left( \frac{2}{\sqrt{\pi}} \int_{\tau(t_l)}^{\tau(t_u)} \exp\left(sy^2\right) dy \right) \\ &= \frac{\sqrt{\pi}}{2\|\Delta \mathbf{v}_{ij}\|} E_{ij}(s, \tau(t_l), \tau(t_u)) \end{aligned} \tag{13}$$

where $E_{ij}(s, \tau(t_l), \tau(t_u))$ is given by

$$E_{ij}(s, t_l, t_u) := \begin{cases} \text{erf}\big(\tau(t_u)\big) - \text{erf}\big(\tau(t_l)\big) & s = +1 \\ \text{erfi}\big(\tau(t_u)\big) - \text{erfi}\big(\tau(t_l)\big) & s = -1 \end{cases}$$

for $\tau(t) := \|\Delta \mathbf{v}_{ij}\|t + \rho_{ij}$. By combining all the results acquired in Equations (10) to (13), we can conclude that

$$\int_{t_l}^{t_u} \exp\Big(\beta(s) + s\|\mathbf{r}_{ij} - \mathbf{r}_{ij}\|^2\Big)\,dt$$

$$= \frac{\sqrt{\pi}}{2\|\Delta \mathbf{v}_{ij}\|} \exp\Big(\beta(s) + s\|\Delta \mathbf{x}_{ij}\|^2 - s\rho_{ij}^2\Big) E_{ij}(s, \tau(t_l), \tau(t_u))$$

□

# B Inference

Our objective function defined by the log-likelihood given in Eq. (4) together with the log-prior regularization term is a non-convex function, so the learning strategy applied for inferring the model's hyper-parameters is crucial to avoid poor-quality local minima in the resulting representations.

The position vectors ($\mathbf{x}$) are initialized uniformly within the $[-1, 1]$ range at random. The bias terms ($\boldsymbol{\beta}$) and velocities ($\mathbf{v}$) are sampled from the standard normal distribution. The prior parameters, ($\boldsymbol{\sigma}_B$, $\boldsymbol{\sigma}_N$) are set to $\mathbf{1}_B/B$ and $\mathbf{1}_N/N$ at the beginning. We follow a sequential learning strategy for training the model, i.e., we optimize different sets of parameters in stages. Firstly, we optimize the velocities ($\mathbf{v}$) for $100$ epochs. Then, we include the initial positions ($\mathbf{x}$) into the optimization procedure, and we continue to train the model by optimizing these two parameters ($\mathbf{x}$, $\mathbf{v}$) together for another $100$ epochs. Finally, we incorporate the bias and prior parameters and optimize all model hyper-parameters together. In total, we use $300$ epochs for the whole learning procedure, and the *Adam optimizer* [20] is employed with an initial learning rate of $0.1$. In the experiments, we set the number of bins ($B$) to $100$ to ensure sufficient capacity for tracking nodes in the latent space ($D = 2$).

# C Experimental Evaluation

In this section, we provide further details regarding the experiments and dynamic visualizations.

## C.1 Datasets.

We treat the networks as undirected and employ the finest available temporal granularity for the input timestamps, including measurements at the level of seconds and milliseconds. We tailor the datasets according to the chosen baselines to enable a meaningful comparison. For instance, we transform dynamic networks into static weighted and unweighted networks by aggregating links over time for static baselines. Additionally, we exclude the non-link events for the baselines since they cannot process these data points.

In the experiments, we have considered the following four real networks:

- *Facebook* [38] is a friendship network signifying one user's presence within another's friend list. We considered users having at least 200 links.
- *NeurIPS* [9] was formed through collaborations among authors whose works were presented at the NeurIPS conference covering the years from 1989 to 2001. We focused on authors with at least ten connections, assuming a year-long duration for each work.
- *Contacts* [8] consists of the interactions among individuals within an office building, encompassing a span of nine days in 2013. (vi) *HyperText* [17] was collected during a conference in which participants wore radio badges tracking their face-to-face interactions, covering a period of approximately $2.5$ days.

- *Infectious* [17] is another interaction network collected during an event in Dublin. In the contact datasets (v-vii), each timestamp associated with a link corresponds to a 20-second connection. When multiple link events occur within a two-minute window, we aggregate these events and treat them as a single link duration.

We also generated two synthetic networks to examine the model's predictive behavior in controlled settings. The link and non-link event times of the *Synthetic-$\alpha$* graph are generated from the Sequential Survival process introduced in Subsection 2.1, and the initial embedding locations and velocities are sampled from a multivariate normal distribution as described in [4]. For the *Synthetic-$\beta$* network, we divide the timeline into equal-sized 8 bins and randomly group the nodes into 10 clusters for each bin. Then, we establish connections between nodes within the same cluster with a probability of $0.8$, while nodes belonging to different clusters are linked with a $10^{-2}$ probability. These links stay persistent until the next bin. We provide a concise overview of the networks in Table 4.

Table 4: Network statistics ($|\mathcal{V}|$: Number of nodes, $|\mathcal{E}_{min}|$: Min. number of events that a dyad has, $|\mathcal{E}_{max}|$: Max. number of events that a dyad has, $|\mathcal{E}|$: Total number of events).

| | $\lvert\mathcal{V}\rvert$ | $\lvert\mathcal{E}_{min}\rvert$ | $\lvert\mathcal{E}_{max}\rvert$ | $\lvert\mathcal{E}\rvert$ | Resolution |
|---|---|---|---|---|---|
| *Synthetic-$\alpha$* | 100 | 1 | 18 | 4286 | N/A |
| *Synthetic-$\beta$* | 100 | 2 | 12 | 8300 | N/A |
| *Contacts* | 92 | 1 | 197 | 5313 | Second |
| *HyperText* | 113 | 1 | 133 | 10450 | Second |
| *Infectious* | 410 | 1 | 29 | 9827 | Second |
| *Facebook* | 461 | 1 | 1 | 10222 | Second |
| *NeurIPS* | 327 | 1 | 6 | 1940 | Year |

### C.2 Continuous-time Dynamic Visualization.

We provide the snapshots of the learned embeddings for various timestamps in Figures 3 to 9 and the intermittent time-persistent linkage structures of the networks are depicted in Figure 10.

### C.3 Optimization.

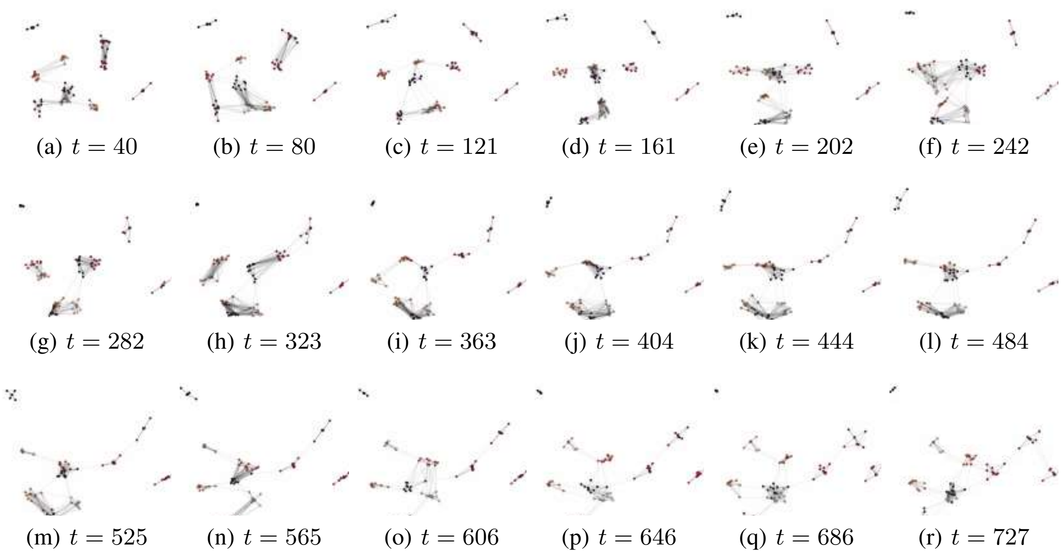
In our experiments, we train all the models for 300 epochs. The number of walks, walk length, and window size parameters are set to 80, 10, and 10 for NODE2VEC and 10, 80, and 10 for CTDNE. The negative sample sizes for HTNE and M$^2$DNE are selected as 10. The learning rate and batch size for LDM, PIVEM, and GRAS$^2$P are set to $0.1$ and $100$, respectively. We tune the scaling factor of the covariance matrix from the set $\{10^1, 10^2, \ldots, 10^{10}\}$, and the number of bins is chosen as 100 for both PIVEM and GRAS$^2$P. For all the other hyperparameters, we employed the default parameters.

### C.4 Weighted-LDM.

In Table 5, we report the performance of the LDM model in terms of the AUC-ROC and AUC-PR scores for the weighted versions of the datasets.

Table 5: Performance of LDM for the weighted versions of the datasets in various prediction tasks.

| | Reconstruction | | Completion | | Future Link Prediction | |
|---|---|---|---|---|---|---|
| | AUC-ROC | AUC-PR | AUC-ROC | AUC-PR | AUC-ROC | AUC-PR |
| *Synthetic-$\alpha$* | $.701 \pm .002$ | $.658 \pm .004$ | $.689 \pm .008$ | $.613 \pm .010$ | $.769 \pm .004$ | $.731 \pm .005$ |
| *Synthetic-$\beta$* | $.561 \pm .013$ | $.549 \pm .015$ | $.485 \pm .013$ | $.487 \pm .007$ | $.497 \pm .016$ | $.500 \pm .016$ |
| *Contacts* | $.585 \pm .003$ | $.538 \pm .004$ | $.505 \pm .010$ | $.490 \pm .009$ | $.818 \pm .003$ | $.764 \pm .005$ |
| *HyperText* | $.533 \pm .005$ | $.501 \pm .007$ | $.519 \pm .017$ | $.496 \pm .009$ | $.605 \pm .011$ | $.554 \pm .013$ |
| *Infectious* | $.693 \pm .008$ | $.612 \pm .011$ | $.686 \pm .004$ | $.616 \pm .008$ | $.938 \pm .005$ | $.903 \pm .014$ |
| *Facebook* | $.681 \pm .005$ | $.615 \pm .005$ | $.714 \pm .005$ | $.656 \pm .008$ | $.726 \pm .005$ | $.678 \pm .005$ |
| *NeurIPS* | $.740 \pm .007$ | $.666 \pm .009$ | $.689 \pm .011$ | $.620 \pm .015$ | $.731 \pm .010$ | $.671 \pm .013$ |

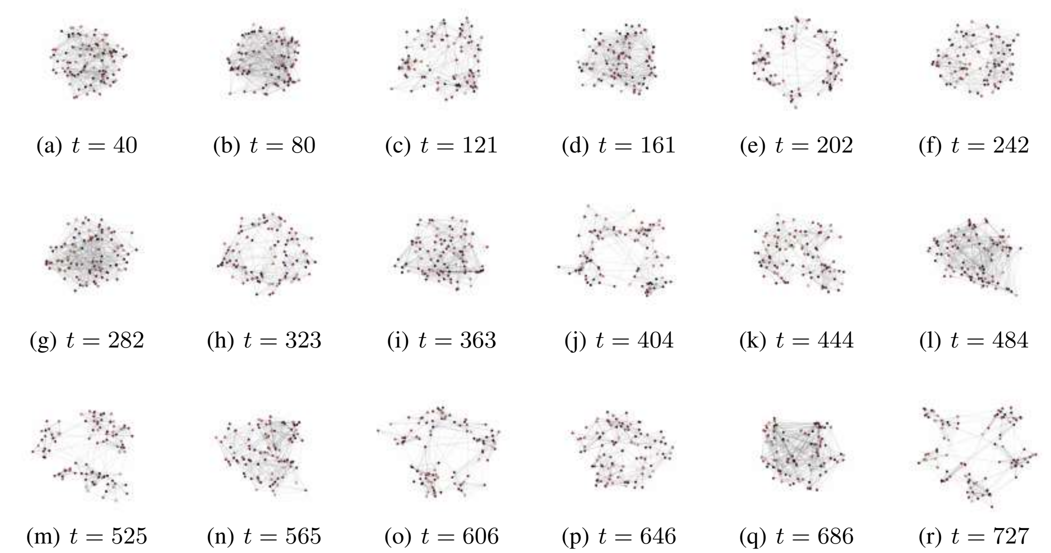


(a) $t = 40$ (b) $t = 80$ (c) $t = 121$ (d) $t = 161$ (e) $t = 202$ (f) $t = 242$

(g) $t = 282$ (h) $t = 323$ (i) $t = 363$ (j) $t = 404$ (k) $t = 444$ (l) $t = 484$

(m) $t = 525$ (n) $t = 565$ (o) $t = 606$ (p) $t = 646$ (q) $t = 686$ (r) $t = 727$

Figure 3: Snapshots of the continuous-time embeddings learned by GRAS$^2$P for various time points over *Synthetic*-$\alpha$.

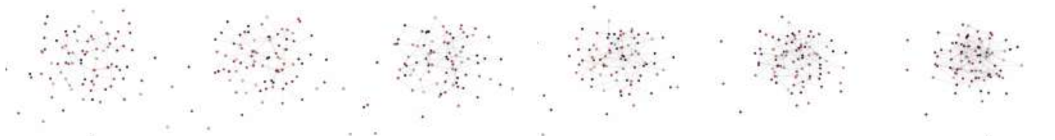


(a) $t = 40$ (b) $t = 80$ (c) $t = 121$ (d) $t = 161$ (e) $t = 202$ (f) $t = 242$

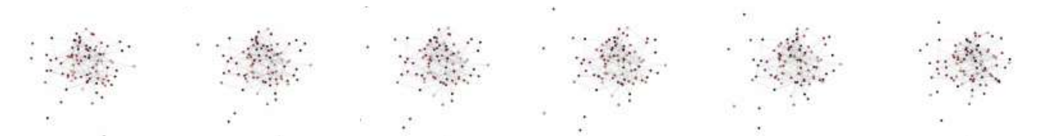
(g) $t = 282$ (h) $t = 323$ (i) $t = 363$ (j) $t = 404$ (k) $t = 444$ (l) $t = 484$

(m) $t = 525$ (n) $t = 565$ (o) $t = 606$ (p) $t = 646$ (q) $t = 686$ (r) $t = 727$

Figure 4: Snapshots of the continuous-time embeddings learned by GRAS$^2$P for various time points over *Synthetic*-$\beta$.



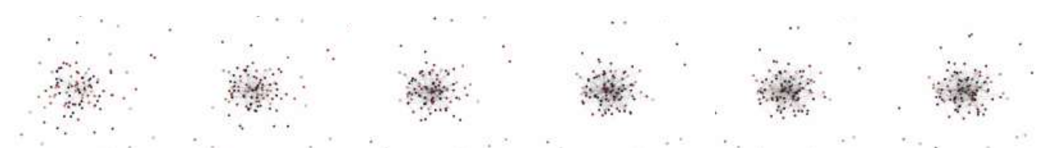
(a) $t = 78678$ (b) $t = 128557$ (c) $t = 178436$ (d) $t = 228315$ (e) $t = 278193$ (f) $t = 328072$



(g) $t = 377951$ (h) $t = 427830$ (i) $t = 477709$ (j) $t = 527587$ (k) $t = 577466$ (l) $t = 627345$

(m) $t = 677224$ (n) $t = 727103$ (o) $t = 776981$ (p) $t = 826860$ (q) $t = 876739$ (r) $t = 926618$

Figure 5: Snapshots of the continuous-time embeddings learned by $\mathrm{GRAS^2P}$ for various time points over *Contacts*.

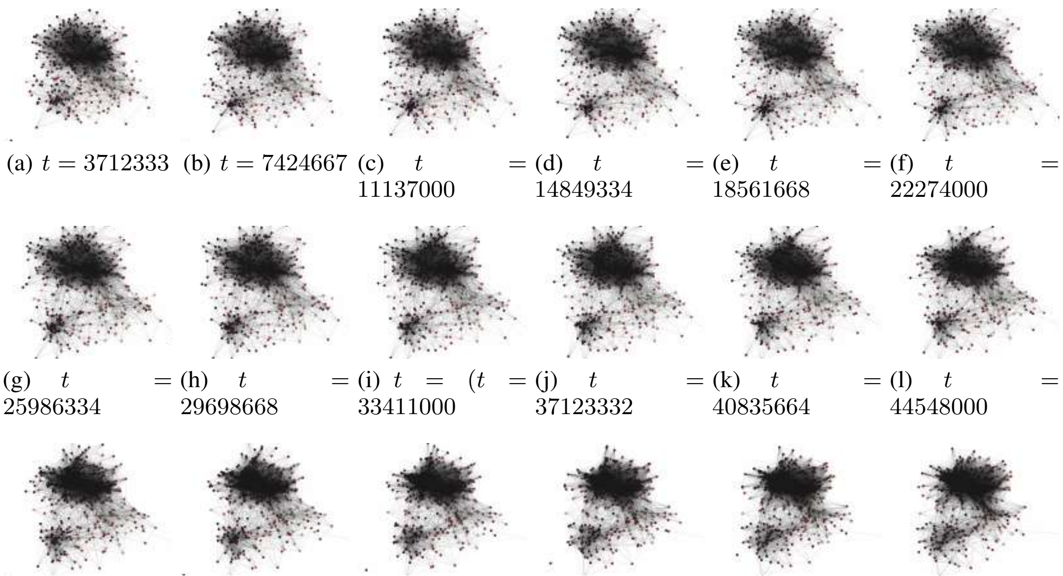


(a) $t = 10719$ (b) $t = 21438$ (c) $t = 32157$ (d) $t = 42876$ (e) $t = 53595$ (f) $t = 64315$

(g) $t = 75034$ (h) $t = 85753$ (i) $t = 96472$ (j) $t = 107191$ (k) $t = 117911$ (l) $t = 128630$

(m) $t = 139349$ (n) $t = 150068$ (o) $t = 160787$ (p) $t = 171507$ (q) $t = 182226$ (r) $t = 192945$

Figure 6: Snapshots of the continuous-time embeddings learned by $\mathrm{GRAS^2P}$ for various time points over *HyperText*.

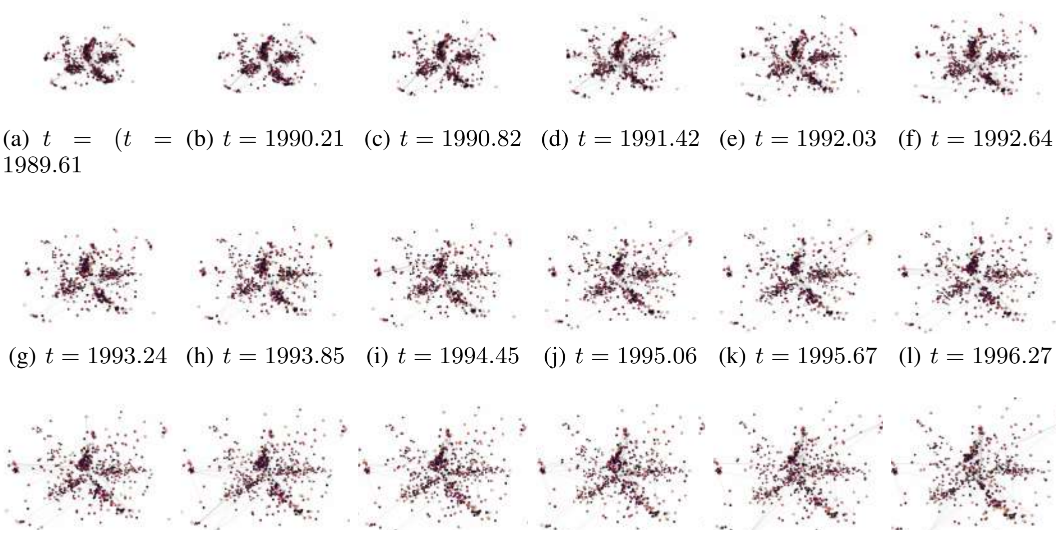


(a) $t = 3712333$ (b) $t = 7424667$ (c) $t = 11137000$ (d) $t = 14849334$ (e) $t = 18561668$ (f) $t = 22274000$

(g) $t = 25986334$ (h) $t = 29698668$ (i) $t = (t = 33411000$ (j) $t = 37123332$ (k) $t = 40835664$ (l) $t = 44548000$

(m) $t = 48260332$ (n) $t = 51972664$ (o) $t = 55685000$ (p) $t = 59397332$ (q) $t = 63109664$ (r) $t = 66822000$

Figure 7: Snapshots of the continuous-time embeddings learned by GRAS$^2$P for various time points over *Facebook*.

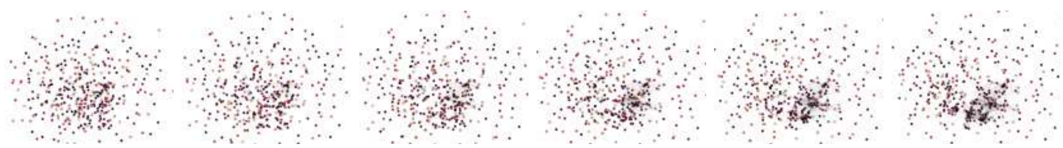


(a) $t = (t = 1989.61$ (b) $t = 1990.21$ (c) $t = 1990.82$ (d) $t = 1991.42$ (e) $t = 1992.03$ (f) $t = 1992.64$

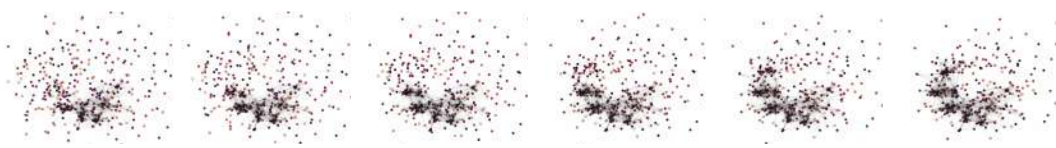
(g) $t = 1993.24$ (h) $t = 1993.85$ (i) $t = 1994.45$ (j) $t = 1995.06$ (k) $t = 1995.67$ (l) $t = 1996.27$

(m) $t = 1996.88$ (n) $t = 1997.48$ (o) $t = 1998.09$ (p) $t = 1998.70$ (q) $t = 1999.30$ (r) $t = 1991.91$

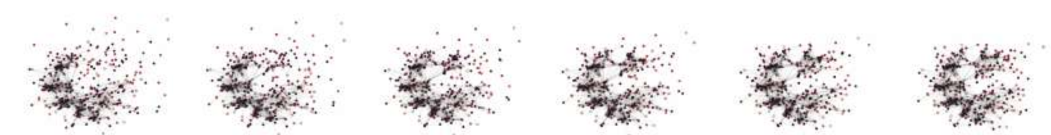
Figure 8: Snapshots of the continuous-time embeddings learned by GRAS$^2$P for various time points over *NeurIPS*.



(a) $t$ = ($t$ = 1435 (b) $t = 2870$ (c) $t = 4306$ (d) $t = 5741$ (e) $t = 7176$ (f) $t = 8612$

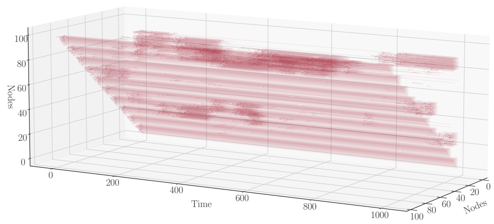


(g) $t = 10047$ (h) $t = 11482$ (i) $t = 12918$ (j) $t = 14353$ (k) $t = 15788$ (l) $t = 17224$

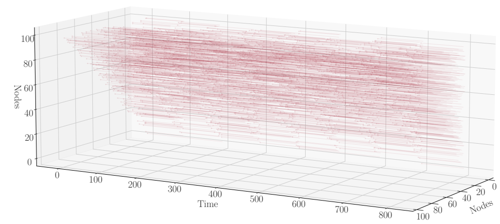


(m) $t = 18659$ (n) $t = 20094$ (o) $t = 21530$ (p) $t = 22965$ (q) $t = 24401$ (r) $t = 25836$

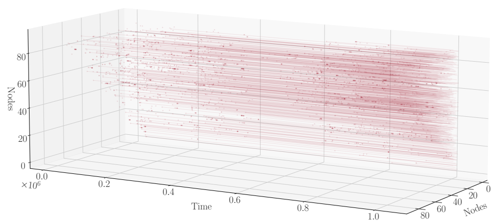
Figure 9: Snapshots of the continuous-time embeddings learned by GRAS$^2$P for various time points over *Infectious*.



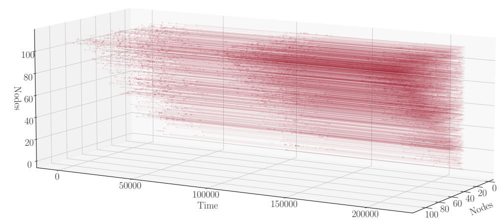
(a) *Synthetic-$\beta$*



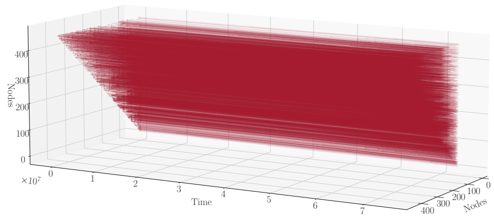
(b) *Synthetic-$\alpha$*



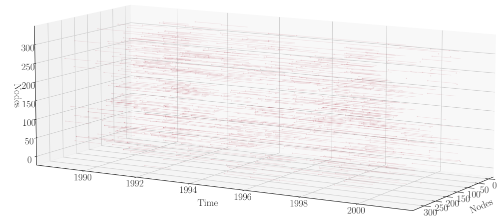
(c) *Contacts*



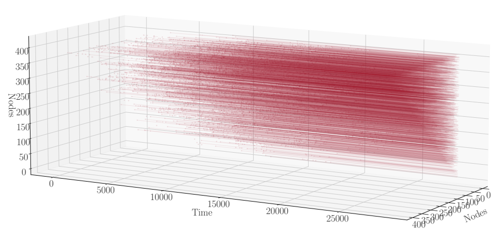
(d) *HyperText*



(e) *Facebook*



(f) *NeurIPS*



(g) *Infectious*

Figure 10: Intermittent time-persistent linkage structures of the networks.

## D Table of Symbols

We provide the list of the symbols used in the manuscript and their explanations in Table 6.

Table 6: Table of symbols

| Symbol | Description |
|---|---|
| $\mathcal{G}$ | Graph |
| $\mathcal{V}$ | Vertex set |
| $\mathcal{E}$ | Edge set |
| $\mathcal{E}_{ij}$ | Edge set of node pair $(i,j)$ |
| $N$ | Number of nodes |
| $D$ | Dimension size |
| $\mathcal{I}_T$ | Time interval |
| $T$ | Time length |
| $B$ | Number of bins |
| $\beta_i$ | Bias term of node $i$ |
| $\mathbf{x}$ | Initial position matrix |
| $\mathbf{v}^{(b)}$ | Velocity matrix for bin $b$ |
| $\mathbf{r}_i(t)$ | Latent representation of node $i$ at time $t$ |
| $\lambda_{ij}(t)$ | Intensity of node pair $(i,j)$ at time $t$ |
| $e_{ij}$ | An event time of node pair $(i,j)$ |
| erf | Error function |
| erfi | Imaginary Error function |

## References


[1] Makan Arastuie, Subhadeep Paul, and Kevin Xu. CHIP: A hawkes process model for continuous-time networks with scalable and consistent estimation. In *NeurIPS*, volume 33, pages 16983–16996, 2020. (Cited on 2)

[2] Ayan Kumar Bhowmick, Koushik Meneni, Maximilien Danisch, Jean-Loup Guillaume, and Bivas Mitra. LouvainNE: Hierarchical louvain method for high quality and scalable network embedding. In *WSDM*, pages 43–51, 2020. (Cited on 1)

[3] Charles Blundell, Jeff Beck, and Katherine A Heller. Modelling reciprocating relationships with hawkes processes. In *NeurIPS*, volume 25, 2012. (Cited on 2)

[4] Abdulkadir Celikkanat, Nikolaos Nakis, and Morten Mørup. Piecewise-velocity model for learning continuous-time dynamic node representations. In *The First Learning on Graphs Conference*, 2022. (Cited on 2, 5, 4)

[5] Yuanfei Dai, Shiping Wang, Neal N Xiong, and Wenzhong Guo. A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics*, 9(5):750, 2020. (Cited on 1)

[6] Sylvain Delattre, Nicolas Fournier, and Marc Hoffmann. Hawkes processes on large networks. *The Annals of Applied Probability*, 26(1):216 – 261, 2016. (Cited on 2)

[7] Xuhui Fan, Bin Li, Feng Zhou, and Scott SIsson. Continuous-time edge modelling using non-parametric point processes. *NeurIPS*, 34:2319–2330, 2021. (Cited on 2)

[8] Mathieu Génois, Christian L Vestergaard, Julie Fournet, André Panisson, Isabelle Bonmarin, and Alain Barrat. Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Network Science*, 3(3):326–347, 2015. (Cited on 3)

[9] Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. *Advances in neural information processing systems*, 17, 2004. (Cited on 3)

[10] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016. (Cited on 1)

[11] William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020. (Cited on 1)

[12] Alan G. Hawkes. Point spectra of some mutually exciting point processes. *J. R. Stat. Soc*, 33(3):438–443, 1971. (Cited on 2)

[13] Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971. (Cited on 2)

[14] Creighton Heaukulani and Zoubin Ghahramani. Dynamic probabilistic models for latent feature propagation in social networks. In *ICML*, pages 275–283, 2013. (Cited on 2)

[15] Tue Herlau, Morten Mørup, and Mikkel Schmidt. Modeling temporal evolution and multiscale structure in networks. In *International Conference on Machine Learning*, pages 960–968. PMLR, 2013. (Cited on 2)

[16] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *JASA*, 97(460):1090–1098, 2002. (Cited on 4)

[17] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1):166–180, 2011. (Cited on 3, 4)

[18] Katsuhiko Ishiguro, Tomoharu Iwata, Naonori Ueda, and Joshua Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. *NeurIPS*, 23, 2010. (Cited on 2)

[19] Stephen P Jenkins. Survival analysis. *Unpublished manuscript, Institute for Social and Economic Research, University of Essex, Colchester, UK*, 42:54–56, 2005. (Cited on 3)

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. (Cited on 3)

[21] Pavel N Krivitsky, Mark S Handcock, Adrian E Raftery, and Peter D Hoff. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social networks*, 31(3):204–213, 2009. (Cited on 5)

[22] Yu Li, Yuan Tian, Jiawei Zhang, and Yi Chang. Learning signed network embedding via graph attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4772–4779, 2020. (Cited on 1)

[23] Yuanfu Lu, Xiao Wang, Chuan Shi, Philip S Yu, and Yanfang Ye. Temporal network embedding with micro-and macro-dynamics. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 469–478, 2019. (Cited on 2)

[24] Nikolaos Nakis, Abdulkadir Celikkanat, Louis Boucherie, Christian Djurhuus, Felix Burmester, Daniel Mathias Holmelund, Monika Frolcová, and Morten Mørup. Characterizing polarization in social networks using the signed relational latent distance model. In *AISTATS*, volume 206, pages 11489–11505. PMLR, 25–27 Apr 2023. (Cited on 1)

[25] Nikolaos Nakis, Abdulkadir Çelikkanat, Sune Lehmann Jørgensen, and Morten Mørup. A hierarchical block distance model for ultra low-dimensional graph representations, 2022. (Cited on 1)

[26] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003. (Cited on 1)

[27] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In *Companion proceedings of the the web conference 2018*, pages 969–976, 2018. (Cited on 2)

[28] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, volume 30, 2017. (Cited on 5)

[29] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, page 701–710, 2014. (Cited on 1)

[30] Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems*, 35:32928–32941, 2022. (Cited on 6)

[31] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. NetSMF: Large-scale network embedding as sparse matrix factorization. In *WWW*, 2019. (Cited on 1)

[32] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and Node2Vec. In *WSDM*, pages 459–467, 2018. (Cited on 1)

[33] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020. (Cited on 2)

[34] Purnamrita Sarkar and Andrew W. Moore. Dynamic social network analysis using latent space models. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1145–1152. MIT Press, 2006. (Cited on 2, 5)

[35] Anna L. Smith, Dena M. Asta, and Catherine A. Calder. The Geometry of Continuous Latent Space Models for Network Data. *Statistical Science*, 34(3):428 – 453, 2019. (Cited on 5)

[36] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015. (Cited on 1)

[37] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. DyRep: Learning representations over dynamic graphs. In *ICLR*, 2019. (Cited on 2)

[38] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 37–42, 2009. (Cited on 3)

[39] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1):4–24, 2021. (Cited on 1)

[40] Guotong Xue, Ming Zhong, Jianxin Li, Jia Chen, Chengshuai Zhai, and Ruochen Kong. Dynamic network embedding survey. *Neurocomputing*, 472:212–223, 2022. (Cited on 2)

[41] Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1975–1985, 2021. (Cited on 2)

[42] D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *IEEE Trans. Big Data*, 6(1), 2020. (Cited on 1)

[43] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2857–2866, 2018. (Cited on 2)