A SUPPLEMENTARY INFORMATION

A.1 LOW-RANK TENSOR DECOMPOSITION

Low-rank tensor decomposition (Kolda & Bader, 2009) aims to factorize a generic tensor into a sum of rank-one tensors. For example, given a 3rd-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$, the rank-$R$ decomposition of $\boldsymbol{\mathcal{X}}$ takes the form of a ternary product between three factor matrices:

$$\boldsymbol{\mathcal{X}} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \equiv \sum_{r=1}^{R} \mathbf{a}_{:r} \otimes \mathbf{b}_{:r} \otimes \mathbf{c}_{:r} \tag{A.1}$$

where $\mathbf{a}_{:r} \in \mathbb{R}^I$, $\mathbf{b}_{:r} \in \mathbb{R}^J$ and $\mathbf{c}_{:r} \in \mathbb{R}^K$ are the columns of the latent factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ and $\otimes$ denotes the outer product. When R is the rank of $\mathcal{X}$, Eq. (A.1) holds with an equality, and the above operation is called Canonical Polyadic (CP) decomposition. Elementwise the previous relation is written as:

$$(\boldsymbol{\mathcal{X}})_{ijk} \approx [\![\mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_k]\!] \equiv \sum_{r=1}^{R} \mathbf{A}_{ir} \mathbf{B}_{jr} \mathbf{C}_{kr} \tag{A.2}$$

where $\mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_k \in \mathbb{R}^R$ are rows of the factor matrices. For 2nd-order tensors (matrices) the operation is equivalent to the low-rank matrix decomposition ($\mathbf{X} \approx \mathbf{A}\mathbf{B}^{\mathrm{T}}$).
For a generic N-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, low-rank decomposition is expressed as:

$$(\boldsymbol{\mathcal{X}})_{i_1 i_2 \ldots i_N} \approx [\![\mathbf{a}^{(1)}_{i_1}, \mathbf{a}^{(2)}_{i_2}, \ldots, \mathbf{a}^{(N)}_{i_N}]\!] \equiv \sum_{r=1}^{R} \mathbf{A}^{(1)}_{i_1 r}\, \mathbf{A}^{(2)}_{i_2 r} \ldots \mathbf{A}^{(N)}_{i_N r} \tag{A.3}$$

where $\mathbf{a}^{(1)}_{i_1}, \mathbf{a}^{(2)}_{i_2}, \ldots, \mathbf{a}^{(N)}_{i_N} \in \mathbb{R}^R$ ($i_n \in \{1, \ldots, I_n\},\ n \in \{1, \ldots, N\}$) are rows of factor matrices $\mathbf{A}^{(1)} \in \mathbb{R}^{I_1 \times R}$, $\mathbf{A}^{(2)} \in \mathbb{R}^{I_2 \times R}, \ldots, \mathbf{A}^{(N)} \in \mathbb{R}^{I_N \times R}$.

A.2 SKIP-GRAM WITH NEGATIVE SAMPLING (SGNS)

The skip-gram approach was initially proposed in WORD2VEC (Mikolov et al., 2013) to obtain low-dimensional representations of words. Starting from a textual corpus of words $w_1, w_2, \ldots, w_m$ from a vocabulary $\mathcal{V}$, it assigns to each word w_s a context corresponding to words $w_{s-T}, \ldots, w_{s-1}, w_{s+1}, \ldots, w_{s+T}$ surrounding w_s in a window of size T. Then a set of training samples $\mathcal{D} = \{(i,j),\ i \in \mathcal{W},\ j \in \mathcal{C}\}$ is built by collecting all the observed word-context pairs, where $\mathcal{W}$ and $\mathcal{C}$ are the vocabularies of words and contexts respectively (normally $\mathcal{W} = \mathcal{C} = \mathcal{V}$). Here we denote as $\#(i,j)$ the number of times (i,j) appears in $\mathcal{D}$. Similarly we use $\#i = \sum_j \#(i,j)$ and $\#j = \sum_i \#(i,j)$ as the number of times each word occurs in $\mathcal{D}$, with relative frequencies $P_{\mathcal{D}}(i,j) = \frac{\#(i,j)}{|\mathcal{D}|}$, $P_{\mathcal{D}}(i) = \frac{\#i}{|\mathcal{D}|}$ and $P_{\mathcal{D}}(j) = \frac{\#j}{|\mathcal{D}|}$.
SGNS computes d-dimensional representations for words and contexts in two matrices $\mathbf{W} \in \mathbb{R}^{|\mathcal{W}| \times d}$ and $\mathbf{C} \in \mathbb{R}^{|\mathcal{C}| \times d}$, performing a binary classification task in which pairs $(i,j) \in \mathcal{D}$ are positive examples and pairs $(i, j_{\mathcal{N}})$ with randomly sampled contexts are negative examples. The probability of the positive class is parametrized as the sigmoid ($\sigma(x) = (1 + e^{-x})^{-1}$) of the inner product of embedding vectors:

$$P[\,(i,j) \in \mathcal{D} \mid \mathbf{w}_i, \mathbf{c}_j\,] = \sigma(\mathbf{w}_i \cdot \mathbf{c}_j) = \sigma\Big(\,(\mathbf{W}\mathbf{C}^{\mathrm{T}})_{ij}\Big) \tag{A.4}$$

and each word-context pair (i,j) contributes to the loss as follows:

$$\ell(i,j) = \log \sigma(\mathbf{w}_i \cdot \mathbf{c}_j) + \sum_{j_{\mathcal{N}} \sim P_{\mathcal{N}}}^{\kappa} \log[1 - \sigma(\mathbf{w}_i \cdot \mathbf{c}_{j_{\mathcal{N}}})] \tag{A.5}$$

$$\simeq \log \sigma(\mathbf{w}_i \cdot \mathbf{c}_j) + \kappa \cdot \mathop{\mathbb{E}}_{j_{\mathcal{N}} \sim P_{\mathcal{N}}} [\log \sigma(-\mathbf{w}_i \cdot \mathbf{c}_{j_{\mathcal{N}}})] \tag{A.6}$$

where the second expression uses the symmetry property $\sigma(-x) = 1 - \sigma(x)$ inside the expected value and κ is the number of negative examples, sampled according to the empirical distribution of

contexts $P_{\mathcal{N}}(j) = P_{\mathcal{D}}(j)$. In the original formulation of WORD2VEC, negative samples are picked from a smoothed distribution $P_{\mathcal{N}}(j) = \frac{(\#j)^{3/4}}{\sum_{j'}(\#j')^{3/4}}$ instead of the unigram probability $\frac{\#j}{|\mathcal{D}|}$, but this smoothing has not been proved to have positive effects in graph representations.
Following results found in Levy & Goldberg (2014), the sum of all $\ell(i,j)$ weighted with the probability each pair (i,j) appears in $\mathcal{D}$ gives the objective function asymptotically optimized:

$$\mathcal{L}^{SGNS} = -\sum_{i=1}^{|\mathcal{W}|}\sum_{j=1}^{|\mathcal{C}|} P_{\mathcal{D}}(i,j)\Big[\log\sigma(\mathbf{w}_i\cdot\mathbf{c}_j) + \kappa\cdot \mathop{\mathbb{E}}_{j_{\mathcal{N}}\sim P_{\mathcal{N}}}\big[\log\sigma(-\mathbf{w}_i\cdot\mathbf{c}_{j_{\mathcal{N}}})\big]\Big] \tag{A.7}$$

$$\dots = -\sum_{i=1}^{|\mathcal{W}|}\sum_{j=1}^{|\mathcal{C}|}\Big[P_{\mathcal{D}}(i,j)\log\sigma(\mathbf{w}_i\cdot\mathbf{c}_j) + \kappa\, P_{\mathcal{N}}(i,j)\log\sigma(-\mathbf{w}_i\cdot\mathbf{c}_j)\Big] \tag{A.8}$$

where $P_{\mathcal{N}}(i,j) = P_{\mathcal{D}}(i)\cdot P_{\mathcal{D}}(j)$ is the probability of (i,j) under assumption of statistical independence.
In Levy & Goldberg (2014) it has been shown that SGNS local loss $\mathcal{L}(i,j)$ exhibits a global optimum with respect to the parameters $\mathbf{w}_i, \mathbf{c}_j$ that satisfies these relations:

$$\frac{\partial\mathcal{L}(i,j)}{\partial(\mathbf{w}_i\cdot\mathbf{c}_j)} = 0 \quad\Leftrightarrow\quad (\mathbf{W}\mathbf{C}^{\mathrm{T}})_{ij} \approx \log\left(\frac{P_{\mathcal{D}}(i,j)}{\kappa\, P_{\mathcal{N}}(i,j)}\right) = \mathrm{PMI}(i,j) - \log(\kappa) \tag{A.9}$$

which tell us that SGNS optimization is equivalent to a rank-d matrix decomposition of the word-context pointwise mutual information (PMI) matrix shifted by a constant. Such factorization is an approximation of the empirical PMI matrix since in the typical case $d \ll \min\big(\{|\mathcal{W}|, |\mathcal{C}|\}\big)$.

A.3 GENERALIZATION OF SGNS TO HIGHER-ORDER REPRESENTATIONS

SGNS can be generalized to learn d-dimensional embeddings from collections of higher-order co-occurrences. Starting with N vocabularies $\big[\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_N\big]$ and a set of N-order tuples $\mathcal{D} = \{(i_1, i_2, \dots, i_N),\ i_1 \in \mathcal{V}_1,\ i_2 \in \mathcal{V}_2,\ \dots,\ i_N \in \mathcal{V}_N\}$, the objective is to learn N factor matrices $\mathbf{A}^{(1)} \in \mathbb{R}^{|\mathcal{V}_1|\times d}, \dots, \mathbf{A}^{(N)} \in \mathbb{R}^{|\mathcal{V}_N|\times d}$ which summarize the co-occurrence statistics of $\mathcal{D}$.

Keeping an example $(i_1, i_2, \dots, i_N) \in \mathcal{D}$, we define the loss with negative sampling scheme fixing i_1 and picking negative tuples $(\nu_2, \dots, \nu_N)$ according to the noise distribution $P_{\mathcal{N}}(\nu_2, \dots, \nu_N) = \prod_{n=2}^{N}\frac{\#\nu_n}{|\mathcal{D}|} \equiv \prod_{n=2}^{N} P_{\mathcal{D}}(\nu_n)$:

$$\begin{aligned}\ell(i_1, i_2, \dots, i_N) = \log\sigma\big([\![\mathbf{a}^{(1)}_{i_1}, \mathbf{a}^{(2)}_{i_2}, \dots, \mathbf{a}^{(N)}_{i_N}]\!]\big) +& \\ + \ \kappa\cdot \mathop{\mathbb{E}}_{\nu_2,\dots,\nu_N\sim P_{\mathcal{N}}}&\Big[\log\sigma\big(-[\![\mathbf{a}^{(1)}_{i_1}, \mathbf{a}^{(2)}_{\nu_2}, \dots, \mathbf{a}^{(N)}_{\nu_N}]\!]\big)\Big]\end{aligned}$$

where each embedding $\mathbf{a}^{(n)}_{i_n}$ is the i_n-th row of the matrix $\mathbf{A}^{(n)}$. The expectation term can be explicited:

$$\mathop{\mathbb{E}}_{\nu_2,\dots,\nu_N\sim P_{\mathcal{N}}}\Big[\log\sigma\big(-[\![\mathbf{a}^{(1)}_{i_1}, \mathbf{a}^{(2)}_{\nu_2}, \dots, \mathbf{a}^{(N)}_{\nu_N}]\!]\big)\Big] = \sum_{j_2,\dots,j_N} P_{\mathcal{N}}(j_2, \dots, j_N)\log\sigma\big(-[\![\mathbf{a}^{(1)}_{i_1}, \mathbf{a}^{(2)}_{j_2}, \dots, \mathbf{a}^{(N)}_{j_N}]\!]\big)$$

Weighting the loss error for each tuple $(i_1, i_2, \dots, i_N)$ with their empirical probability $P_{\mathcal{D}}(i_1, i_2, \dots, i_N) = \frac{\#(i_1, i_2, \dots, i_N)}{|\mathcal{D}|}$, and defining $[\![\mathbf{a}^{(1)}_{i_1}, \mathbf{a}^{(2)}_{i_2}, \dots, \mathbf{a}^{(N)}_{i_N}]\!] \equiv m_{i_1 i_2 \dots i_N}$, we obtain the global objective with the sum over all combinations of vocabulary elements:

$$\begin{aligned}\mathcal{L} &= -\sum_{i_1, i_2, \dots, i_N} P_{\mathcal{D}}(i_1, i_2, \dots, i_N)\Big[\log\sigma(m_{i_1 i_2\dots i_N}) + \kappa\sum_{j_2,\dots,j_N} P_{\mathcal{N}}(j_2, \dots, j_N)\log\sigma(-m_{i_1 j_2 \dots j_N})\Big] \\ &= -\sum_{i_1, i_2, \dots, i_N} P_{\mathcal{D}}(i_1, i_2, \dots, i_N)\log\sigma(m_{i_1 i_2\dots i_N}) + \\ &\qquad - \kappa\sum_{i_1, i_2, \dots, i_N} P_{\mathcal{D}}(i_1, i_2, \dots, i_N)\sum_{j_2,\dots,j_N} P_{\mathcal{N}}(j_2, \dots, j_N)\log\sigma(-m_{i_1 j_2 \dots j_N})\end{aligned}$$

In the second term we can notice that only $P_{\mathcal{D}}(i_1, i_2, \ldots, i_N)$ depends on the $N-1$ indices $(i_2, \ldots, i_N)$, so performing the sum over that subset of indices we obtain the marginal distribution $\sum_{i_2 \ldots i_N} P_{\mathcal{D}}(i_1, i_2, \ldots, i_N) = P_{\mathcal{D}}(i_1)$. Finally renaming indices $\{j_h\} \to \{i_h\}$ and observing that $P_{\mathcal{D}}(i_1) P_{\mathcal{N}}(i_2, \ldots, i_N) \equiv P_{\mathcal{N}}(i_1, i_2, \ldots, i_N)$, we obtain the final loss:

$$\mathcal{L}^{HOSGNS} = -\sum_{i_1,\ldots,i_N} \Big[P_{\mathcal{D}}(i_1, \ldots, i_N) \log \sigma(m_{i_1 \ldots i_N}) + \kappa \cdot P_{\mathcal{N}}(i_1, \ldots, i_N) \log \sigma(-m_{i_1 \ldots i_N}) \Big] \tag{A.10}$$

In particular for the 3rd-order and 4th-order cases, with vocabularies $\mathcal{V}_1 = \mathcal{W},\ \mathcal{V}_2 = \mathcal{C},\ \mathcal{V}_3 = \mathcal{T},\ \mathcal{V}_4 = \mathcal{S}$ and embedding matrices $\mathbf{A}^{(1)} = \mathbf{W},\ \mathbf{A}^{(2)} = \mathbf{C},\ \mathbf{A}^{(3)} = \mathbf{T},\ \mathbf{A}^{(4)} = \mathbf{S}$, we have the loss functions minimized by our time-varying graph embedding model:

$$\mathcal{L}^{(3rd)} = -\sum_{i,j,k} \Big[P_{\mathcal{D}}(i,j,k) \log \sigma\big(\llbracket \mathbf{w}_i, \mathbf{c}_j, \mathbf{t}_k \rrbracket\big) + \kappa\, P_{\mathcal{N}}(i,j,k) \log \sigma\big(-\llbracket \mathbf{w}_i, \mathbf{c}_j, \mathbf{t}_k \rrbracket\big) \Big]$$

$$\mathcal{L}^{(4th)} = -\sum_{i,j,k,l} \Big[P_{\mathcal{D}}(i,j,k,l) \log \sigma\big(\llbracket \mathbf{w}_i, \mathbf{c}_j, \mathbf{t}_k, \mathbf{s}_l \rrbracket\big) + \kappa\, P_{\mathcal{N}}(i,j,k,l) \log \sigma\big(-\llbracket \mathbf{w}_i, \mathbf{c}_j, \mathbf{t}_k, \mathbf{s}_l \rrbracket\big) \Big]$$

A.4 HOSGNS AS IMPLICIT TENSOR FACTORIZATION

Here we show the equivalence of HOSGNS to low-rank tensor factorization of the shifted PMI tensor into factor matrices, which is a straightforward generalization of previous proofs done for SGNS.

Theorem. Let $\mathcal{D} = \{(i_1, i_2, \ldots, i_N),\ i_1 \in \mathcal{V}_1,\ i_2 \in \mathcal{V}_2,\ \ldots,\ i_N \in \mathcal{V}_N\}$ a training set of higher-order co-occurrences and $\mathrm{PMI}(i_1, \ldots, i_N) = \log\left(\frac{P_{\mathcal{D}}(i_1,\ldots,i_N)}{P_{\mathcal{N}}(i_1,\ldots,i_N)}\right)$ the entries of the pointwise mutual information tensor computed from $\mathcal{D}$. Let $\mathbf{A}^{(1)} \in \mathbb{R}^{|\mathcal{V}_1| \times d}, \ldots, \mathbf{A}^{(N)} \in \mathbb{R}^{|\mathcal{V}_N| \times d}$ embedding matrices of HOSGNS. For d sufficiently large, HOSGNS has the same global optimum as the canonical polyadic decomposition of SPMI_κ, the PMI tensor shifted by $\log \kappa$.

Proof. We consider each relation $\llbracket \mathbf{a}^{(1)}_{i_1}, \ldots, \mathbf{a}^{(N)}_{i_N} \rrbracket \equiv m_{i_1 \ldots i_n}$ as a mapping from combinations of embedding vectors to elements of a tensor $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{|\mathcal{V}_1| \times \cdots \times |\mathcal{V}_N|}$. The global loss $\mathcal{L} = \sum_{i_1 \ldots i_N} \mathcal{L}(i_1, \ldots, i_N)$ in Eq. (A.10) is the sum of local losses computed from elements of $\boldsymbol{\mathcal{M}}$:

$$\mathcal{L}(i_1, \ldots, i_N) = -\big[P_{\mathcal{D}}(i_1, \ldots, i_N) \log \sigma(m_{i_1 \ldots i_N}) + \kappa\, P_{\mathcal{N}}(i_1, \ldots, i_N) \log \sigma(-m_{i_1 \ldots i_N}) \big]$$

For sufficiently large d (i.e. allowing for a perfect reconstruction of SPMI_κ), each $m_{i_1 \ldots i_N}$ can assume a value independently of the others, and we can treat the loss function $\mathcal{L}$ as a sum of independent addends, restricting the optimization problem to looking at the local objective and its derivative respect to $m_{i_1 \ldots i_N}$:

$$\begin{aligned} \frac{\partial \mathcal{L}(i_1, \ldots, i_N)}{\partial m_{i_1 \ldots i_N}} &= \kappa\, P_{\mathcal{N}}(i_1, \ldots, i_N) \sigma(m_{i_1 \ldots i_N}) - P_{\mathcal{D}}(i_1, \ldots, i_N) \big[1 - \sigma(m_{i_1 \ldots i_N}) \big] \\ &= \big[P_{\mathcal{D}}(i_1, \ldots, i_N) + \kappa\, P_{\mathcal{N}}(i_1, \ldots, i_N) \big] \sigma(m_{i_1 \ldots i_N}) - P_{\mathcal{D}}(i_1, \ldots, i_N) \end{aligned}$$

where we have used $\frac{d\sigma}{dx} = \sigma(x)(1 - \sigma(x))$. To compare the derivative with zero, we use the identities $P_{\mathcal{D}} = (P_{\mathcal{D}} + \kappa\, P_{\mathcal{N}})(1 + \frac{\kappa\, P_{\mathcal{N}}}{P_{\mathcal{D}}})^{-1}$ and $(1+x)^{-1} = \sigma(\log x^{-1})$:

$$\frac{\partial \mathcal{L}(i_1, \ldots, i_N)}{\partial m_{i_1 \ldots i_N}} = \left[P_{\mathcal{D}}(i_1, \ldots, i_N) + \kappa\, P_{\mathcal{N}}(i_1, \ldots, i_N) \right] \left[\sigma(m_{i_1 \ldots i_N}) - \sigma\left(\log \frac{P_{\mathcal{D}}(i_1, \ldots, i_N)}{\kappa\, P_{\mathcal{N}}(i_1, \ldots, i_N)} \right) \right]$$

from which it follows that the derivative is 0 when elements $m_{i_1 \ldots i_N}$ are equal to the shifted PMI tensor entries:

$$\frac{\partial \mathcal{L}(i_1, \ldots, i_N)}{\partial m_{i_1 \ldots i_N}} = 0 \quad \Leftrightarrow \quad \sum_{r=1}^{d} \mathbf{A}^{(1)}_{i_1 r} \ldots \mathbf{A}^{(N)}_{i_N r} = \log\left(\frac{P_{\mathcal{D}}(i_1, \ldots, i_N)}{\kappa\, P_{\mathcal{N}}(i_1, \ldots, i_N)} \right) = \mathrm{SPMI}_\kappa(i_1, \ldots, i_N) \tag{A.11}$$

Since we have assumed that d is large enough to ensure an exact reconstruction of SPMI_κ, and this is true if $d \approx R = \mathrm{rank}(\mathrm{SPMI}_\kappa)$, Eq. (A.11) is consistent with the canonical polyadic decomposition of the shifted PMI tensor. □

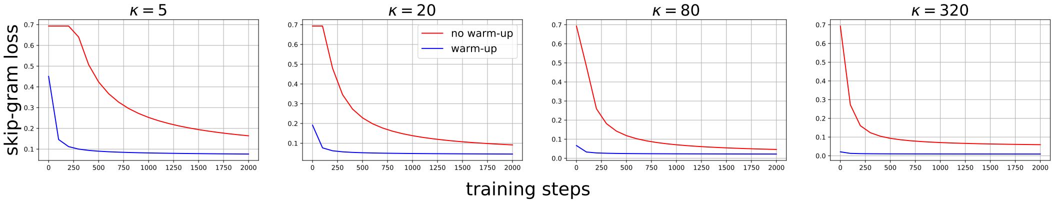


Figure 1: Impact of additional warm-up steps in the decrease of HOSGNS$^{(dyn)}$ loss function $\mathcal{L}^{(\text{bce})}$ respect to the number of training iterations, for LyonSchool dataset and with different negative sampling sizes κ. The loss function is normalized with a factor $(\kappa+1)$ for accounting the different contribute of the negative sampling parameter in $\mathcal{L}^{(\text{bce})}$.

Remark. In the typical case when $d \ll R$, the tensor reconstruction of Eq. (A.11) is not exact, since the tensor is compressed into lower-dimensional factor matrices, but it still holds as a low-rank approximation:

$$\sum_{r=1}^{d \ll R} \mathbf{A}^{(1)}_{i_1 r} \dots \mathbf{A}^{(N)}_{i_N r} \approx \text{SPMI}_\kappa(i_1, \dots, i_N)$$

A.5 DESCRIPTION OF THE WARM-UP PROCEDURE

Warm-up is usually referred as the sequence of weights updates, at the beginning of training, performed in order to reduce over-fitting at early stages, and especially it is useful when data samples are highly differentiated (Goyal et al., 2017; Devlin et al., 2018). Here we designed the warm-up strategy with a different aim, i.e. finding an advantageous configuration of model parameters to initialize trainable weights. In particular we show that we can preliminarily optimize embedding vectors in order to ensure that all higher-order products $m_{ijk\dots} = [\![\mathbf{w}_i, \mathbf{c}_j, \mathbf{t}_k \dots]\!]$ return the same quantity m, regardless of the indices combination $(i, j, k \dots)$. The value m can be chosen in order to make the cross entropy error as minimum as possible before passing empirical data samples to the model.
We start with a random initialization where embedding weights are realizations of random variables i.i.d. according to a normal distribution:

$$\mathbf{W}_{ir}, \mathbf{C}_{jr}, \mathbf{T}_{kr} \dots \sim \mathcal{N}(0, d^{-2}) \ , \ r = 1 \dots d$$

Once chosen m we can fix Hadamard products optimizing a squared error loss function:

$$\mathcal{L}^{(\text{warmup})} = \sum_{ijk\dots} \Big([\![\mathbf{w}_i, \mathbf{c}_j, \mathbf{t}_k \dots]\!] - m \Big)^2$$

The optimal value of m is stated by the following theorem:

Theorem. Assuming the same value for each higher-order inner product in the set:

$$\mathcal{S} = \{ m_{ijk\dots}, i \in \mathcal{W}, \ j \in \mathcal{C}, \ k \in \mathcal{T}, \dots \}$$

the cross entropy error of HOSGNS (Equation (3.8) of the main paper) is minimum when every $m_{ijk\dots} \equiv m = -\log \kappa$.

Proof. Given the hypothesis, the objective function to minimize takes the form:

$$\begin{aligned} \mathcal{L}^{(\text{bce})} &= -\frac{1}{B} \Big[\sum_{(ijk\dots) \sim P_\mathcal{D}}^{B} \log \sigma(m) + \kappa \sum_{(ijk\dots) \sim P_\mathcal{N}}^{B} \log \sigma(-m) \Big] \\ &= -[\log \sigma(m) + \kappa \log \sigma(-m)] \equiv \ell(m) \end{aligned}$$

where m is the returned value for each $m_{ijk\dots} \in \mathcal{S}$. Solving the equation $\frac{dl}{dm} = 0$ we get:

$$\frac{1}{\sigma(m)} \frac{d\sigma}{dx}\Big|_{x=m} - \kappa \frac{1}{\sigma(-m)} \frac{d\sigma}{dx}\Big|_{x=-m} = 0 \Rightarrow \kappa = \frac{\sigma(-m)}{\sigma(m)} = e^{-m} \Rightarrow m = -\log \kappa$$

□

In Figure 1 is shown the effectiveness of the addition of extra warm-up steps in loss optimization.

Table 1: Number of class components for each label in LyonSchool dataset.

Class name	Number of children or teachers
CP-A	23
CP-B	25
CE1-A	23
CE1-B	26
CE2-A	23
CE2-B	22
CM1-A	21
CM1-B	23
CM2-A	22
CM2-B	24
Teachers	10

A.6 DESCRIPTION OF PARAMETER SETTINGS

Unless otherwise declared, all the embeddings are trained with a dimension d of 128 for node classification and 192 for temporal event reconstruction.

HOSGNS variants were optimized with Adam (Kingma & Ba, 2014) fixing the negative samples weight $\kappa = 5$, the sample size $B = 50000$ and linearly decaying the learning rate from a starting value of 0.05 for 4000 iterations. For $\boldsymbol{\mathcal{A}}^{(dyn)}$ we set the random walks context window $T = 10$. Before training we apply 100 warm-up steps with uniform sampling of 10^5 terms per iteration in the squared loss. Models are implemented in Tensorflow[1].

For **DyANE** , as in the original paper, we optimized node2vec[2] with default hyperparameters ($p = q = 1$, the same value κ=5 for negative samples and the same context window size $T = 10$ that we chose for HOSGNS). The number of SGD epochs is 1 since we did not observe any improvement in downstream tasks by increasing the number of epochs.

For **DynGEM**, with the code made available online by the authors[3], we trained the model with SGD with momentum (learning rate 10^{-3} and momentum coefficient 0.99) for 100 iterations in the first time-step and 30 for the others. We set the internal layer sizes of the autoencoder to $[400, 250, d]$.

DynamicTriad is trained with Adagrad (learning rate 10^{-1}) with 100 epochs and negative/positive samples ratio set to 5. Coefficients β_0 and β_1 related to social homofily and temporal smoothness are seto to 0.1. We used the reference implementation available in the official repository[4].

We tested a few combinations of other hyperparameters, and reported the results with the ones described above, since we observed that the improvement is minimal and does not invalidate the results. Due to the stochastic nature of the training, each of the above embedding models is trained 5 times for more robust performance estimates in downstream tasks.

All the experiments are executed on a 64 bit Ubuntu 18.04.4 LTS system with Intel(R) Core(TM) i7-5930K CPU, 6 cores, 3.50GHz clock frequency, 64 GB RAM, and two Nvidia GeForce GTX Titan X, each with 12 GB memory.

A.7 EMBEDDING SPACE VISUALIZATION

One of the main advantages of HOSGNS is that it able to disentangle the role of nodes and time by learning representations of nodes and time intervals separately. In this section, we include plots with two-dimensional projections of these embeddings, made with UMAP (McInnes et al., 2018) for manifold learning and non-linear dimensionality reduction. With these plots, we show that the embedding matrices learned by HOSGNS$^{(stat)}$ and HOSGNS$^{(dyn)}$ approaches successfully capture both the structure and the dynamics of the time-varying graph.

---

[1] `https://github.com/tensorflow/tensorflow`

[2] `https://github.com/snap-stanford/snap/tree/master/examples/node2vec`

[3] `http://www-scf.usc.edu/~nkamra/`

[4] `https://github.com/luckiezhou/DynamicTriad`

Temporal information can be represented by associating each embedding vector to its corresponding $k \in \mathcal{T}$, while graph structure can be represented by associating each embedding vector to a community membership. While community membership can be estimated by different community detection methods, we choose to use a dataset with ground truth data containing node membership information. We consider in this section the LYONSCHOOL dataset as a case study, widely investigated in literature respect to structural and spreading properties (Stehlé et al., 2011; Barrat & Cattuto, 2013; Starnini et al., 2012; Panisson et al., 2013; Sapienza et al., 2018; Galimberti et al., 2018). This dataset includes metadata (Table 1) concerning the class of each participant of the school (10 different labels for children and 1 label for teachers), and we identify the community membership of each individual according to these labels (*class* labels). Moreover we also assign *time* labels according to activation of individual nodes in temporal snapshots. To show how disentangled representations capture different aspects of the evolving graph, in Figure 2 we plot individual representations of nodes $i \in \mathcal{V}$ and time slices $k \in \mathcal{T}$ labeled according to the class membership and the time snapshot respectively. In Figure 3 we visualize representations of temporal nodes $i^{(k)} \in \mathcal{V}^{(\mathcal{T})}$, computed as Hadamard products of nodes and time embeddings, in order to highlight both structural and dynamical aspects captured by the same set of embedding vectors. In Figure 4 we see dynamic node embeddings computed with baseline methods without dissociating structure and time.

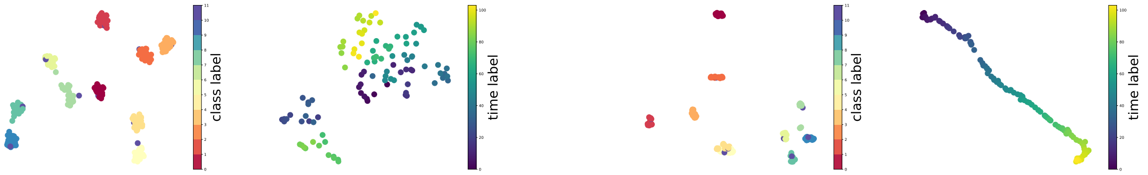
A.8 INTRINSIC AND EXTRINSIC EVALUATION OF EMBEDDING REPRESENTATIONS

Here we report results on empirical datasets about intrinsic evaluation of the quality of embedding learned with HOSGNS, besides to completing the extrinsic evaluation in downstream tasks already reported (partially) in the main paper.

As intrinsic evaluation, in Figure 5 we probe the capability of the model to reconstruct the shifted PMI tensor entries computing the higher order product of embedding vectors, operation optimized during the training phase to classify non-zero elements of the tensor itself. We verify the goodness of fit estimating the square of the Pearson coefficient between the distribution of actual PMI values and the estimated ones, having fixed the model $\kappa = 5$ during training.

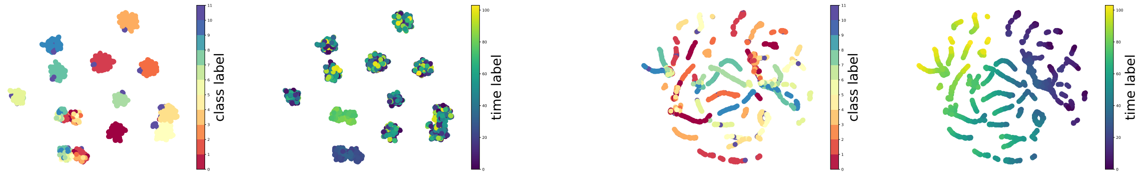
In Tables 3, 4 and 5 we report Macro-F1 scores in downstream tasks, as extrinsic evaluation, with different operations used to construct embeddings for the logistic regression. For both node classification and temporal event reconstruction, in Table 2 we present definitions of different operators employed (Hadamard included, the only one displayed in the paper). For node classification, we show in Tables 3 and 4 results related to all tested combinations of epidemic parameters (β, μ) used to simulate SIR processes.

In Figures 6 and 7 we report a sensitivity analysis with the effect of the embedding size d, the negative sampling constant κ and the number of training steps E on prediction performances in downstream tasks.



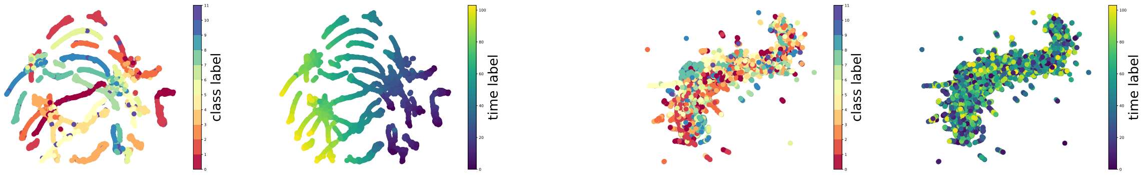
(a) HOSGNS$^{(stat)}$ (b) HOSGNS$^{(dyn)}$

Figure 2: Two-dimensional projections of the 128-dim embedding manifold spanned by embedding matrices $\mathbf{W}$ (left of each panel) and $\mathbf{T}$ (right of each panel), trained on LyonSchool data, of HOSGNS model trained on: (a) $\boldsymbol{\mathcal{P}}^{(stat)}$ and (b) $\boldsymbol{\mathcal{P}}^{(dyn)}$. These plots show how the community structure and the evolution of time is captured by individual node embeddings $\{\mathbf{w}_i\}_{i\in\mathcal{V}}$ and time embeddings $\{\mathbf{t}_k\}_{k\in\mathcal{T}}$.

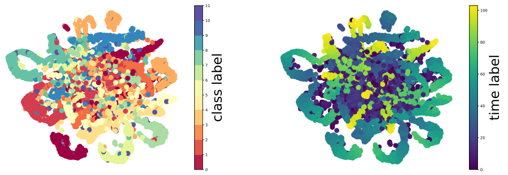


(a) HOSGNS$^{(stat)}$ (b) HOSGNS$^{(dyn)}$

Figure 3: Two-dimensional projections of the 128-dim embedding manifold spanned by dynamic node embeddings, trained on LyonSchool data and obtained with Hadamard products $\{\mathbf{w}_i \circ \mathbf{t}_k\}_{(i,k)\in\mathcal{V}^{(\mathcal{T})}}$ between rows of $\mathbf{W}$ (node embeddings) and $\mathbf{T}$ (time embeddings), from HOSGNS model trained on: (a) $\boldsymbol{\mathcal{P}}^{(stat)}$ and (b) $\boldsymbol{\mathcal{P}}^{(dyn)}$. We highlight the temporal participation to communities (left of each panel) and the time interval of activation (right of each panel).

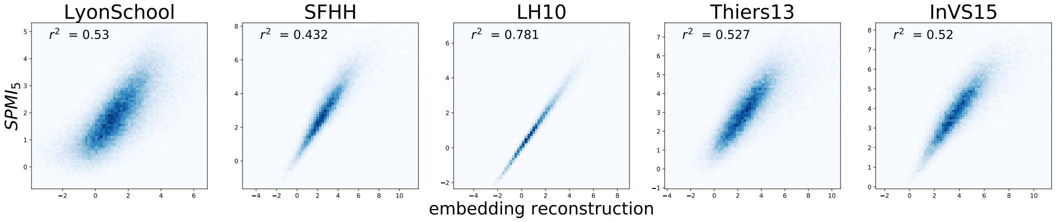


(a) DyANE (b) DynGEM

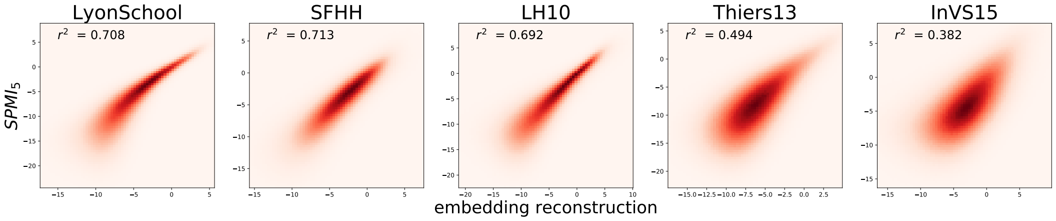


(c) DynamicTriad

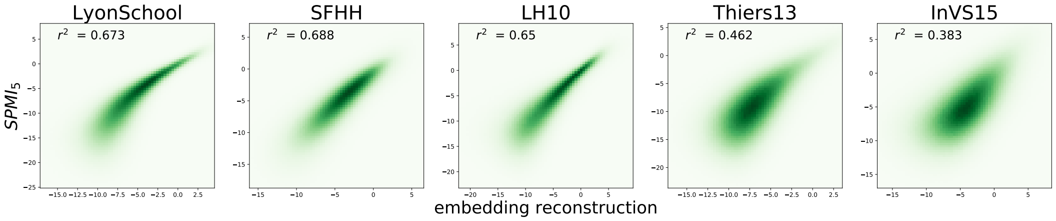
Figure 4: Two-dimensional projections of the 128-dim embedding manifold spanned by dynamic node embeddings for LyonSchool data learned with: (a) DyANE, (b) DynGEM and (c) DynamicTriad. As in Figure 3 we highlight the temporal participation to communities (left of each panel) and the time interval of activation (right of each panel).



(a) HOSGNS$^{(stat)}$



(b) HOSGNS$^{(dyn)}$

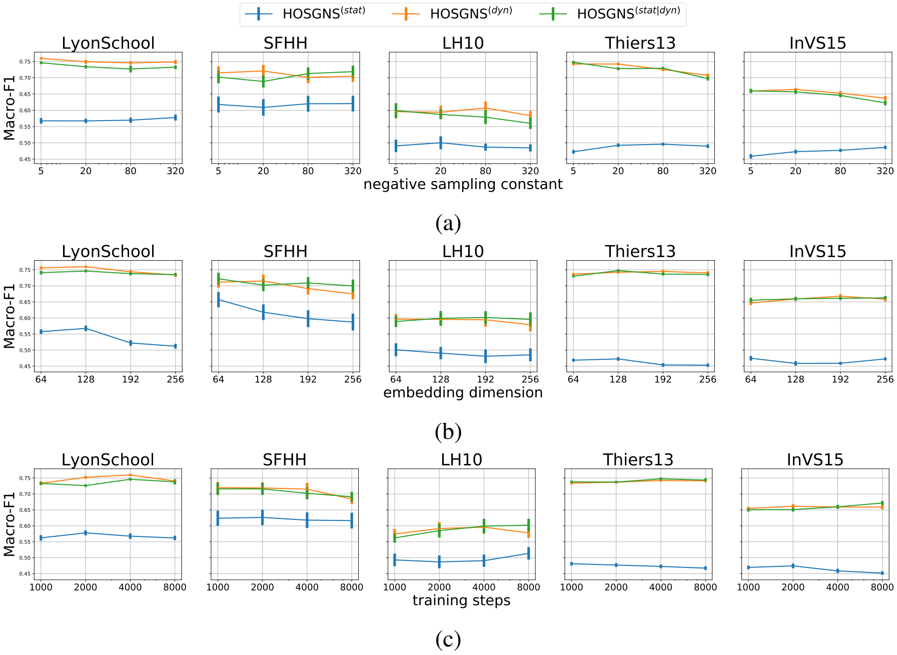


(c) HOSGNS$^{(stat|dyn)}$

Figure 5: 2D histograms of shifted PMI values $\mathrm{SPMI}_5(i, j, k \ldots)$ (whereas are greater than $-\infty$) versus embedding reconstruction from higher-order inner products $[\![\mathbf{w}_i, \mathbf{c}_j, \mathbf{t}_k, \ldots]\!]$, with HOSGNS models trained on: (a) $\boldsymbol{\mathcal{P}}^{(stat)}$, (b) $\boldsymbol{\mathcal{P}}^{(dyn)}$ and (c) $\boldsymbol{\mathcal{P}}^{(stat|dyn)}$. The histograms were built by uniformly sampling 10^7 entries from the SPMI_5 tensors.

Table 2: Operators and their definitions used to combine different embeddings learned with HOSGNS for tensors of order 3 (HOSGNS$^{(stat)}$) and 4 (HOSGNS$^{(dyn)}$ and HOSGNS$^{(stat|dyn)}$), applied to temporal node $i^{(k)}$ in node classification and to link (i, j, k) in temporal event reconstruction. All operations, except *Concat*, are described element-wise.

Operator	SGNS order	Node Classification	Temp. Event Reconstruction
Average	$3rd, 4th$	$\frac{1}{2}(\mathbf{w}_i + \mathbf{t}_k)$	$\frac{1}{3}(\mathbf{w}_i + \mathbf{c}_j + \mathbf{t}_k)$
Hadamard	$3rd$ $4th$	$\mathbf{w}_i \circ \mathbf{t}_k$	$\mathbf{w}_i \circ \mathbf{c}_j \circ \mathbf{t}_k$ $\mathbf{w}_i \circ \mathbf{c}_j \circ \mathbf{t}_k \circ \mathbf{s}_k$
Weighted-L1	$3rd, 4th$	$\lvert\mathbf{w}_i - \mathbf{t}_k\rvert$	$\frac{1}{3}(\lvert\mathbf{w}_i - \mathbf{t}_k\rvert + \lvert\mathbf{w}_i - \mathbf{c}_j\rvert + \lvert\mathbf{c}_j - \mathbf{t}_k\rvert)$
Weighted-L2	$3rd, 4th$	$(\mathbf{w}_i - \mathbf{t}_k)^2$	$\frac{1}{3}[(\mathbf{w}_i - \mathbf{t}_k)^2 + (\mathbf{w}_i - \mathbf{c}_j)^2 + (\mathbf{c}_j - \mathbf{t}_k)^2]$
Concat	$3rd, 4th$	$[\mathbf{w}_i, \mathbf{t}_k]$	$[\mathbf{w}_i, \mathbf{c}_j, \mathbf{t}_k]$

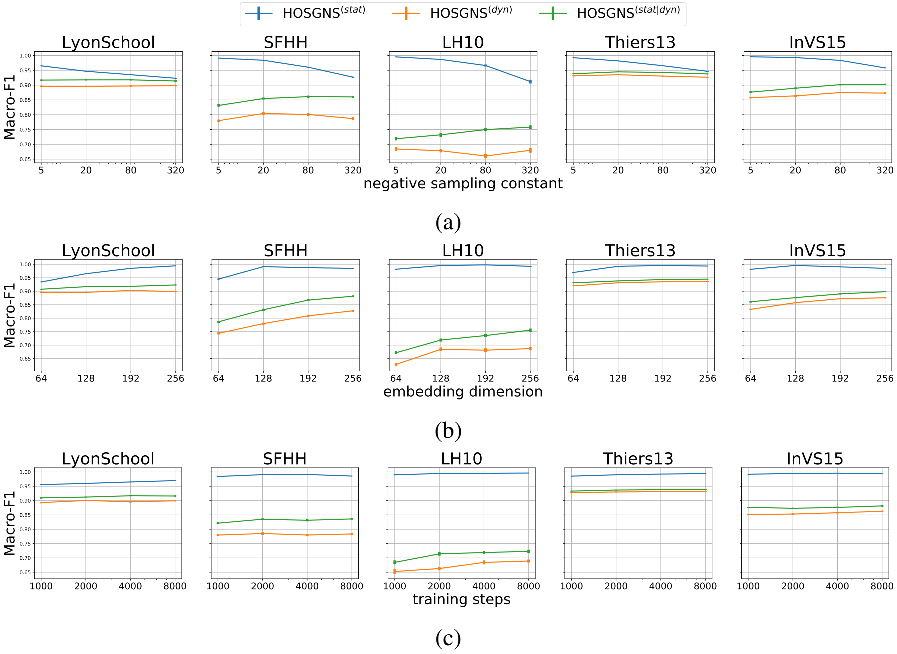


(a)

(b)

(c)

Figure 6: Macro-F1 scores related to classification of nodes in SIR states from simulations with epidemic parameters $(\beta, \mu) = (0.125, 0.001)$, computed (a) varying the negative sampling parameter κ, (b) varying the embedding dimension and (c) varying the number of training iterations E. In each panel remaining parameters are fixed to $d = 128$, $\kappa = 5$ and $E = 4000$. Time-resolved embedding vectors of nodes are computed with Hadamard product as explained in Table 2.



(a)

(b)

(c)

Figure 7: Macro-F1 scores related to temporal event reconstruction, computed (a) varying the negative sampling parameter κ, (b) varying the embedding dimension and (c) varying the number of training iterations E. In each panel remaining parameters are fixed to $d = 128$, $\kappa = 5$ and $E = 4000$. Time-resolved embedding vectors of edges are computed with Hadamard product as explained in Table 2.

Table 3: Macro-F1 scores for classification of nodes in epidemic states according to a SIR process with parameters (β, μ) previously shown in the paper. Here for each HOSGNS variant we tested different operators to produce node-time representations, all with a dimension $d = 128$, used as input to a Logistic Regression. For each (β, μ) we highlight the two highest scores and underline the best one.

(β, μ)	Model	Operator			Dataset		
			LYONSCHOOL	SFHH	LH10	THIERS13	INVS15
$(0.25, 0.002)$	DYANE	-	$\mathbf{78.1 \pm 0.5}$	67.0 ± 1.2	52.5 ± 1.7	71.9 ± 0.6	$\mathbf{64.3 \pm 0.8}$
	DYNGEM		58.7 ± 2.8	35.9 ± 1.1	34.5 ± 0.7	35.5 ± 1.2	58.8 ± 1.1
	DYNAMICTRIAD		31.0 ± 0.4	28.8 ± 0.4	29.9 ± 0.3	30.3 ± 0.2	30.4 ± 0.2
	HOSGNS$^{(stat)}$	Average	56.3 ± 0.9	55.5 ± 1.3	51.1 ± 1.1	58.0 ± 0.5	53.0 ± 0.8
		Hadamard	55.5 ± 0.8	57.3 ± 1.1	45.9 ± 0.9	46.9 ± 0.7	44.5 ± 0.7
		Weighted-L1	53.9 ± 0.9	49.2 ± 0.9	49.5 ± 1.2	46.7 ± 0.6	45.5 ± 0.8
		Weighted-L2	53.2 ± 0.7	47.5 ± 0.9	48.8 ± 1.0	46.8 ± 0.6	45.0 ± 0.7
		Concat	69.4 ± 0.9	59.8 ± 1.4	54.4 ± 1.2	61.2 ± 0.8	56.6 ± 0.8
	HOSGNS$^{(dyn)}$	Average	73.1 ± 0.5	66.5 ± 1.1	62.5 ± 1.5	69.5 ± 0.9	62.4 ± 0.8
		Hadamard	$\underline{\mathbf{79.2}} \pm \mathbf{0.5}$	$\underline{\mathbf{69.1}} \pm \mathbf{1.1}$	59.6 ± 1.5	71.8 ± 1.2	$\underline{\mathbf{64.6}} \pm \mathbf{0.7}$
		Weighted-L1	75.7 ± 0.5	66.2 ± 1.2	59.0 ± 1.0	70.8 ± 0.7	61.9 ± 0.8
		Weighted-L2	74.9 ± 0.6	67.0 ± 1.1	60.5 ± 1.2	72.2 ± 0.5	62.5 ± 0.6
		Concat	77.1 ± 0.5	$\mathbf{68.8 \pm 1.0}$	$\underline{\mathbf{63.5}} \pm \mathbf{1.5}$	$\underline{\mathbf{72.9}} \pm \mathbf{0.6}$	63.1 ± 0.8
	HOSGNS$^{(stat\|dyn)}$	Average	72.3 ± 0.5	65.2 ± 0.9	61.0 ± 1.3	69.5 ± 0.7	62.5 ± 1.0
		Hadamard	77.4 ± 0.6	67.4 ± 1.2	59.7 ± 1.2	$\mathbf{72.5 \pm 0.7}$	64.2 ± 1.0
		Weighted-L1	73.4 ± 0.8	66.7 ± 1.1	57.2 ± 1.3	70.1 ± 0.8	63.1 ± 0.9
		Weighted-L2	73.4 ± 0.6	65.0 ± 1.2	57.8 ± 1.2	70.0 ± 0.7	63.5 ± 0.9
		Concat	76.1 ± 0.6	67.9 ± 1.1	$\mathbf{62.6 \pm 1.5}$	70.9 ± 0.6	62.0 ± 0.8
$(0.125, 0.001)$	DYANE	-	$\mathbf{75.3 \pm 0.4}$	$\underline{\mathbf{71.6}} \pm \mathbf{1.9}$	59.0 ± 1.8	72.4 ± 0.3	65.8 ± 0.6
	DYNGEM		58.9 ± 2.9	37.0 ± 4.1	41.0 ± 1.4	32.5 ± 1.2	59.0 ± 1.2
	DYNAMICTRIAD		31.2 ± 0.5	35.0 ± 3.3	30.5 ± 0.7	27.4 ± 0.3	29.5 ± 0.2
	HOSGNS$^{(stat)}$	Average	54.9 ± 0.9	59.4 ± 2.6	50.4 ± 2.1	59.8 ± 0.5	55.5 ± 0.6
		Hadamard	56.8 ± 0.9	61.8 ± 2.4	49.1 ± 1.9	47.3 ± 0.6	45.9 ± 0.7
		Weighted-L1	55.5 ± 0.7	54.5 ± 2.9	49.7 ± 2.0	49.8 ± 0.6	46.8 ± 0.6
		Weighted-L2	52.6 ± 0.8	53.0 ± 3.0	47.9 ± 2.1	47.7 ± 0.5	45.3 ± 0.6
		Concat	66.6 ± 1.2	65.9 ± 2.2	52.2 ± 1.9	63.8 ± 0.5	58.4 ± 0.6
	HOSGNS$^{(dyn)}$	Average	68.0 ± 1.2	68.5 ± 2.1	59.0 ± 2.1	71.0 ± 0.7	65.2 ± 0.7
		Hadamard	$\underline{\mathbf{76.0}} \pm \mathbf{0.4}$	$\mathbf{71.5 \pm 2.0}$	59.6 ± 2.0	$\mathbf{74.2 \pm 0.4}$	$\mathbf{65.9 \pm 0.6}$
		Weighted-L1	73.2 ± 0.5	69.2 ± 2.0	58.7 ± 1.7	72.6 ± 0.5	65.6 ± 0.5
		Weighted-L2	71.2 ± 0.7	69.4 ± 2.0	59.1 ± 2.2	73.2 ± 0.4	65.2 ± 0.5
		Concat	73.1 ± 0.5	71.3 ± 1.9	57.3 ± 2.0	72.7 ± 0.5	65.5 ± 0.6
	HOSGNS$^{(stat\|dyn)}$	Average	68.0 ± 0.7	68.5 ± 2.1	58.8 ± 2.0	70.7 ± 0.5	64.6 ± 0.4
		Hadamard	74.6 ± 0.4	70.2 ± 1.9	$\underline{\mathbf{59.9}} \pm \mathbf{2.3}$	$\underline{\mathbf{74.8}} \pm \mathbf{0.4}$	$\underline{\mathbf{66.0}} \pm \mathbf{0.6}$
		Weighted-L1	71.8 ± 0.5	69.7 ± 2.1	58.8 ± 2.3	72.2 ± 0.5	64.8 ± 0.5
		Weighted-L2	70.8 ± 0.6	69.9 ± 2.0	58.4 ± 2.2	72.6 ± 0.5	64.7 ± 0.5
		Concat	71.8 ± 0.6	70.7 ± 1.9	$\mathbf{59.7 \pm 2.3}$	72.1 ± 0.5	65.2 ± 0.7
$(0.0625, 0.002)$	DYANE	-	72.2 ± 0.6	64.9 ± 1.7	59.0 ± 1.2	68.0 ± 0.5	$\underline{\mathbf{60.2}} \pm \mathbf{0.5}$
	DYNGEM		56.4 ± 2.7	35.9 ± 4.1	35.8 ± 1.2	32.9 ± 1.2	55.0 ± 0.6
	DYNAMICTRIAD		29.5 ± 0.5	33.1 ± 2.5	29.6 ± 0.4	27.4 ± 0.3	28.4 ± 0.2
	HOSGNS$^{(stat)}$	Average	54.7 ± 0.8	59.0 ± 2.6	51.2 ± 1.1	56.7 ± 0.5	52.1 ± 0.5
		Hadamard	55.5 ± 0.7	57.6 ± 2.2	49.4 ± 0.8	45.5 ± 0.4	43.6 ± 0.5
		Weighted-L1	53.7 ± 0.7	52.7 ± 2.9	49.7 ± 1.1	46.2 ± 0.5	43.9 ± 0.5
		Weighted-L2	54.0 ± 0.8	51.4 ± 3.0	48.0 ± 0.9	45.5 ± 0.4	43.0 ± 0.5
		Concat	65.5 ± 0.7	61.3 ± 2.6	55.3 ± 1.4	61.9 ± 0.5	53.3 ± 0.6
	HOSGNS$^{(dyn)}$	Average	67.8 ± 0.7	66.6 ± 2.1	$\mathbf{63.0 \pm 1.2}$	67.6 ± 0.4	59.1 ± 0.6
		Hadamard	$\underline{\mathbf{73.5}} \pm \mathbf{0.5}$	65.7 ± 1.6	61.1 ± 1.2	$\underline{\mathbf{69.5}} \pm \mathbf{0.3}$	59.6 ± 0.5
		Weighted-L1	70.8 ± 0.6	65.9 ± 1.9	62.4 ± 1.3	67.5 ± 0.4	58.0 ± 0.6
		Weighted-L2	71.4 ± 0.5	66.5 ± 2.2	59.0 ± 1.1	68.5 ± 0.5	57.5 ± 0.7
		Concat	72.0 ± 0.7	$\underline{\mathbf{68.3}} \pm \mathbf{2.1}$	$\underline{\mathbf{64.0}} \pm \mathbf{1.5}$	68.2 ± 0.4	59.9 ± 0.6
	HOSGNS$^{(stat\|dyn)}$	Average	67.8 ± 0.5	64.3 ± 1.9	60.1 ± 1.2	66.7 ± 0.4	$\underline{\mathbf{60.2}} \pm \mathbf{0.5}$
		Hadamard	$\mathbf{72.9 \pm 0.6}$	66.3 ± 1.9	58.2 ± 1.1	$\mathbf{68.5 \pm 0.4}$	59.0 ± 0.7
		Weighted-L1	70.3 ± 0.5	$\mathbf{67.2 \pm 2.2}$	59.2 ± 1.2	67.2 ± 0.5	58.0 ± 0.5
		Weighted-L2	70.0 ± 0.5	66.6 ± 2.2	59.5 ± 1.3	66.7 ± 0.5	57.6 ± 0.6
		Concat	71.3 ± 0.6	66.8 ± 2.2	62.8 ± 1.6	68.3 ± 0.3	59.2 ± 0.6

Table 4: Macro-F1 scores for classification of nodes in epidemic states according to a SIR process with other combinations (β, μ) not shown in the paper. Here for each HOSGNS variant we tested different operators to produce node-time representations, all with a dimension $d = 128$, used as input to a Logistic Regression. For each (β, μ) we highlight the two highest scores and underline the best one. In the case $(\beta, \mu) = (0.125, 0.004)$ results for datasets LH10 and InVS15 are discarded since the SIR simulation does not meet the condition $|I_{|\mathcal{T}|/2}| \geq 1$, as explained in DyANE.

(β, μ)	Model	Operator			Dataset		
			LYONSCHOOL	SFHH	LH10	THIERS13	INVS15
$(0.125, 0.002)$	DYANE	-	$\mathbf{77.1 \pm 0.4}$	$\mathbf{68.4 \pm 0.9}$	54.8 ± 1.5	71.6 ± 0.4	62.4 ± 0.5
	DYNGEM		57.1 ± 2.7	32.8 ± 1.3	35.0 ± 0.8	34.4 ± 1.0	57.1 ± 0.7
	DYNAMICTRIAD		30.4 ± 0.4	29.3 ± 0.4	30.1 ± 0.3	29.0 ± 0.3	29.4 ± 0.2
	HOSGNS$^{(stat)}$	Average	57.5 ± 0.8	54.0 ± 0.9	48.9 ± 1.0	58.0 ± 0.7	53.7 ± 0.6
		Hadamard	55.4 ± 0.9	55.9 ± 0.8	44.9 ± 1.0	46.3 ± 0.4	44.8 ± 0.6
		Weighted-L1	53.8 ± 0.8	48.5 ± 0.8	48.2 ± 1.0	46.9 ± 0.5	45.8 ± 0.6
		Weighted-L2	52.6 ± 0.8	46.5 ± 0.8	46.9 ± 1.0	45.2 ± 0.5	44.0 ± 0.7
		Concat	69.5 ± 0.5	59.0 ± 1.0	53.8 ± 1.3	62.1 ± 0.8	56.2 ± 0.6
	HOSGNS$^{(dyn)}$	Average	72.1 ± 0.6	66.0 ± 0.9	$\mathbf{60.8 \pm 1.2}$	70.1 ± 0.5	61.2 ± 0.7
		Hadamard	$\underline{\mathbf{77.5}} \pm \mathbf{0.5}$	$\underline{\mathbf{68.8}} \pm \mathbf{0.8}$	58.7 ± 1.1	$\underline{\mathbf{72.6}} \pm \mathbf{0.5}$	$\mathbf{63.3 \pm 0.6}$
		Weighted-L1	73.5 ± 0.6	66.4 ± 0.8	59.6 ± 1.5	70.7 ± 0.5	60.2 ± 0.5
		Weighted-L2	73.8 ± 0.5	66.1 ± 1.0	58.6 ± 1.2	71.0 ± 0.4	61.0 ± 0.7
		Concat	74.8 ± 0.5	67.5 ± 0.8	$\underline{\mathbf{62.2}} \pm \mathbf{1.5}$	71.0 ± 0.4	62.9 ± 0.5
	HOSGNS$^{(stat\|dyn)}$	Average	71.2 ± 0.8	65.8 ± 0.8	59.4 ± 1.0	69.6 ± 0.5	62.0 ± 0.6
		Hadamard	75.2 ± 0.6	68.1 ± 0.8	59.7 ± 1.1	$\mathbf{72.0 \pm 0.5}$	$\underline{\mathbf{63.4}} \pm \mathbf{0.6}$
		Weighted-L1	73.0 ± 0.5	64.7 ± 0.8	57.3 ± 1.2	70.0 ± 0.5	61.0 ± 0.6
		Weighted-L2	72.0 ± 0.6	63.8 ± 0.9	57.0 ± 0.9	70.1 ± 0.6	62.4 ± 0.6
		Concat	73.7 ± 0.6	66.5 ± 0.8	60.1 ± 1.3	70.3 ± 0.5	61.6 ± 0.8
$(0.1875, 0.001)$	DYANE	-	$\mathbf{74.7 \pm 0.7}$	67.7 ± 1.2	$\underline{\mathbf{63.4}} \pm \mathbf{1.8}$	72.7 ± 0.4	$\underline{\mathbf{68.6}} \pm \mathbf{0.4}$
	DYNGEM		57.4 ± 2.8	36.2 ± 2.6	41.4 ± 1.3	34.8 ± 1.3	61.2 ± 0.9
	DYNAMICTRIAD		32.3 ± 0.5	31.5 ± 0.8	30.5 ± 0.4	27.9 ± 0.3	30.0 ± 0.2
	HOSGNS$^{(stat)}$	Average	56.4 ± 0.8	57.6 ± 1.7	50.5 ± 1.4	58.4 ± 0.8	56.9 ± 0.8
		Hadamard	56.9 ± 0.8	59.4 ± 1.7	48.5 ± 1.1	49.0 ± 0.6	46.2 ± 0.8
		Weighted-L1	53.5 ± 0.9	51.3 ± 1.8	47.3 ± 0.8	48.2 ± 0.5	47.7 ± 0.6
		Weighted-L2	52.4 ± 0.9	48.9 ± 1.9	47.1 ± 1.1	48.5 ± 0.6	47.2 ± 0.7
		Concat	67.3 ± 0.7	62.7 ± 1.5	51.6 ± 1.6	63.7 ± 0.8	59.5 ± 0.8
	HOSGNS$^{(dyn)}$	Average	69.9 ± 0.5	66.3 ± 1.5	$\mathbf{62.6 \pm 1.9}$	71.0 ± 0.7	65.4 ± 0.8
		Hadamard	$\underline{\mathbf{76.5}} \pm \mathbf{0.4}$	68.6 ± 1.1	62.4 ± 1.7	$\underline{\mathbf{74.8}} \pm \mathbf{0.5}$	$\mathbf{67.9 \pm 0.7}$
		Weighted-L1	72.1 ± 0.5	68.3 ± 1.5	62.4 ± 1.9	72.5 ± 0.6	64.9 ± 0.7
		Weighted-L2	71.5 ± 0.5	67.7 ± 1.4	60.7 ± 1.9	72.5 ± 0.6	66.4 ± 0.7
		Concat	73.0 ± 0.5	$\mathbf{69.1 \pm 1.4}$	59.6 ± 2.1	72.9 ± 0.6	65.1 ± 0.7
	HOSGNS$^{(stat\|dyn)}$	Average	69.6 ± 0.7	66.2 ± 1.4	61.6 ± 1.8	69.6 ± 0.7	66.1 ± 0.6
		Hadamard	74.5 ± 0.4	$\underline{\mathbf{69.4}} \pm \mathbf{1.4}$	62.5 ± 2.0	$\mathbf{73.6 \pm 0.6}$	67.3 ± 0.5
		Weighted-L1	71.1 ± 0.6	68.5 ± 1.4	58.6 ± 1.8	71.0 ± 0.6	66.0 ± 0.8
		Weighted-L2	71.0 ± 0.6	67.1 ± 1.5	59.0 ± 1.6	71.0 ± 0.7	65.8 ± 0.5
		Concat	73.0 ± 0.6	68.2 ± 1.4	60.1 ± 1.8	72.1 ± 0.7	65.1 ± 0.8
$(0.125, 0.004)$	DYANE	-	76.0 ± 0.5	63.0 ± 0.9	-	67.7 ± 0.6	-
	DYNGEM		57.9 ± 2.5	34.0 ± 0.8		35.0 ± 1.1	
	DYNAMICTRIAD		31.2 ± 0.3	29.7 ± 0.4		29.5 ± 0.2	
	HOSGNS$^{(stat)}$	Average	56.5 ± 0.6	52.6 ± 0.8	-	54.2 ± 0.7	-
		Hadamard	54.5 ± 0.8	54.2 ± 0.8		44.3 ± 0.5	
		Weighted-L1	54.0 ± 0.8	46.9 ± 0.8		44.7 ± 0.6	
		Weighted-L2	52.0 ± 0.9	45.6 ± 0.7		44.2 ± 0.6	
		Concat	68.2 ± 1.2	57.4 ± 1.1		58.1 ± 0.9	
	HOSGNS$^{(dyn)}$	Average	73.3 ± 0.6	62.9 ± 0.9	-	66.0 ± 0.7	-
		Hadamard	$\underline{\mathbf{77.2}} \pm \mathbf{0.4}$	63.5 ± 1.0		$\underline{\mathbf{68.6}} \pm \mathbf{0.6}$	
		Weighted-L1	75.0 ± 0.6	62.6 ± 0.8		67.1 ± 0.5	
		Weighted-L2	73.2 ± 1.2	62.3 ± 0.9		67.2 ± 0.5	
		Concat	$\mathbf{77.0 \pm 0.4}$	$\underline{\mathbf{64.8}} \pm \mathbf{0.8}$		67.8 ± 0.7	
	HOSGNS$^{(stat\|dyn)}$	Average	72.0 ± 0.6	60.3 ± 0.8	-	65.6 ± 0.6	-
		Hadamard	74.4 ± 0.7	$\mathbf{64.5 \pm 1.0}$		$\mathbf{68.1 \pm 0.6}$	
		Weighted-L1	72.3 ± 0.7	61.1 ± 0.9		65.4 ± 0.4	
		Weighted-L2	72.8 ± 0.6	60.3 ± 0.8		66.6 ± 0.5	
		Concat	75.2 ± 0.4	63.1 ± 1.0		67.4 ± 0.6	

Table 5: Macro-F1 scores for temporal event reconstruction. Here for each HOSGNS variant we tested different operators to produce link-time representations, all with a dimension $d = 192$, used as input to a Logistic Regression. We highlight in bold the best two overall scores for each dataset. For baseline models we underline their highest score.

Model	Operator			Dataset		
		LYONSCHOOL	SFHH	LH10	THIERS13	INVS15
DYANE	Average	56.4 ± 0.4	52.9 ± 0.5	52.3 ± 0.6	51.0 ± 0.4	52.7 ± 0.4
	Hadamard	89.7 ± 0.3	$\underline{86.5} \pm 0.3$	$\underline{74.6} \pm 0.6$	94.7 ± 0.1	94.1 ± 0.1
	Weighted-L1	90.2 ± 0.2	83.3 ± 0.5	73.3 ± 0.7	94.7 ± 0.1	94.4 ± 0.2
	Weighted-L2	$\underline{90.6} \pm 0.2$	84.5 ± 0.5	72.0 ± 0.5	$\underline{\mathbf{95.0}} \pm 0.1$	$\underline{\mathbf{94.8}} \pm \mathbf{0.2}$
	Concat	65.7 ± 0.4	53.8 ± 0.4	56.2 ± 0.6	57.0 ± 0.4	50.9 ± 0.4
DYNGEM	Average	57.7 ± 0.5	56.8 ± 0.7	$\underline{54.8} \pm 1.5$	40.4 ± 1.5	42.8 ± 0.9
	Hadamard	$\underline{62.2} \pm 0.4$	55.1 ± 1.0	52.5 ± 1.6	40.8 ± 1.5	43.7 ± 1.0
	Weighted-L1	58.4 ± 0.6	52.3 ± 0.7	50.9 ± 1.2	$\underline{41.3} \pm 1.6$	44.8 ± 0.9
	Weighted-L2	53.7 ± 0.6	47.0 ± 0.8	47.0 ± 1.3	39.2 ± 1.2	43.6 ± 0.6
	Concat	60.4 ± 0.4	$\underline{57.8} \pm 0.3$	48.9 ± 1.7	36.9 ± 1.3	$\underline{45.7} \pm 1.0$
DYNAMICTRIAD	Average	51.7 ± 0.2	56.9 ± 0.4	60.2 ± 0.6	58.1 ± 0.2	56.1 ± 0.3
	Hadamard	60.3 ± 0.3	58.9 ± 0.4	59.5 ± 0.5	62.2 ± 0.3	64.7 ± 0.3
	Weighted-L1	$\underline{79.1} \pm 0.4$	72.3 ± 0.4	75.5 ± 0.6	70.8 ± 0.3	78.1 ± 0.2
	Weighted-L2	77.4 ± 0.4	$\underline{73.4} \pm 0.4$	$\underline{\mathbf{77.4}} \pm 0.5$	$\underline{72.4} \pm 0.2$	$\underline{78.9} \pm 0.3$
	Concat	52.2 ± 0.2	53.4 ± 0.3	55.9 ± 0.7	55.1 ± 0.2	53.2 ± 0.3
HOSGNS$^{(stat)}$	Average	61.2 ± 0.4	53.2 ± 0.4	53.0 ± 0.6	56.0 ± 0.4	51.3 ± 0.4
	Hadamard	$\mathbf{98.5 \pm 0.1}$	$\mathbf{98.8 \pm 0.1}$	$\mathbf{99.8 \pm 0.1}$	$\mathbf{99.6 \pm 0.1}$	$\mathbf{99.1 \pm 0.1}$
	Weighted-L1	67.2 ± 0.4	60.6 ± 0.5	57.4 ± 0.6	66.7 ± 0.6	59.7 ± 0.4
	Weighted-L2	68.5 ± 0.3	60.6 ± 0.5	55.6 ± 0.6	68.0 ± 0.4	58.8 ± 0.5
	Concat	63.3 ± 0.5	54.5 ± 0.4	52.2 ± 1.0	58.7 ± 0.7	50.5 ± 0.5
HOSGNS$^{(dyn)}$	Average	63.4 ± 0.4	53.9 ± 0.4	50.3 ± 0.9	57.2 ± 0.5	50.7 ± 0.5
	Hadamard	90.3 ± 0.2	80.9 ± 0.4	68.1 ± 0.7	93.5 ± 0.2	87.2 ± 0.2
	Weighted-L1	80.5 ± 0.4	63.2 ± 0.4	56.6 ± 0.9	82.1 ± 0.4	66.5 ± 0.4
	Weighted-L2	80.4 ± 0.4	63.7 ± 0.4	56.4 ± 0.6	82.1 ± 0.3	62.9 ± 0.4
	Concat	64.1 ± 0.4	53.9 ± 0.4	50.9 ± 0.9	58.2 ± 0.7	50.9 ± 0.5
HOSGNS$^{(stat\|dyn)}$	Average	63.4 ± 0.4	54.2 ± 0.5	52.6 ± 0.8	56.8 ± 0.6	50.4 ± 0.5
	Hadamard	$\mathbf{91.8 \pm 0.2}$	$\mathbf{86.7 \pm 0.4}$	73.6 ± 0.6	94.3 ± 0.1	89.0 ± 0.2
	Weighted-L1	81.5 ± 0.3	64.0 ± 0.4	58.1 ± 0.8	83.7 ± 0.3	66.8 ± 0.5
	Weighted-L2	81.2 ± 0.3	64.6 ± 0.5	55.4 ± 0.6	82.7 ± 0.4	63.5 ± 0.3
	Concat	61.5 ± 0.4	53.0 ± 0.4	52.9 ± 0.9	58.3 ± 0.6	49.5 ± 0.6

A.9 ABLATION STUDY ON DOWNSTREAM TASKS

We performed additional ablation analysis in order to quantify the different contribute of structural and temporal representations learned by the proposed model in downstream tasks. With this aim, we executed node classification and event reconstruction in the same experimental setting used to evaluate prediction scores showed in Tables 3,4 and 5, but using as predictors node or time embeddings alone.

For node features, we assigned in node classification embedding vectors $\mathbf{w}_i$ to each active node $i^{(k)}$ and embedding vectors $\mathbf{w}_i \circ \mathbf{c}_j$ to a given temporal link (i, j, k) for event reconstruction. For time features, we used $\mathbf{t}_k$ in node classification and $\mathbf{t}_k \circ \mathbf{s}_k$ in event reconstruction with HOSGNS$^{(dyn)}$ and HOSGNS$^{(stat|dyn)}$; for HOSGNS$^{(stat)}$ we used embedding $\mathbf{t}_k$ as well.

We compared results with two baselines respectively for node-related and time-related tasks:

- Node embedding learned with DEEPWALK (Perozzi et al., 2014) applied to the static network obtained joining all temporal snapshots in a single graph.
- Time embedding computed using Positional Encoding (Vaswani et al., 2017) to obtain sets of representations aware of temporal order.

In Tables 6 and 7 we report performance scores related to the two downstream tasks for different sets of features used as predictors. Results for node classification of epidemic states show that HOSGNS node features have almost always better performances respect to DEEPWALK static features, moreover when HOSGNS time features are compared with Positional Encoding we observe comparable or better results with the latter method. Scores for temporal event reconstruction display a similar tendency.

Table 6: Macro-F1 scores deriving from the ablation study for classification of nodes in epidemic states according to a SIR process with parameters (β, μ). Node and time representations, all with a dimension $d = 128$, are separately used as input to a Logistic Regression. For each (β, μ) we highlight the highest score for the two types of predictors.

(β, μ)	Predictors	Model			Dataset		
			LYONSCHOOL	SFHH	LH10	THIERS13	INVS15
$(0.25, 0.002)$	Node Embeddings	DEEPWALK	32.2 ± 0.5	32.6 ± 0.5	31.1 ± 1.2	31.8 ± 0.3	32.8 ± 0.5
		HOSGNS$^{(stat)}$	$\mathbf{32.9 \pm 0.5}$	36.1 ± 0.5	46.7 ± 1.9	$\mathbf{33.1 \pm 0.3}$	33.2 ± 0.5
		HOSGNS$^{(dyn)}$	32.7 ± 0.5	$\mathbf{36.4 \pm 0.5}$	$\mathbf{50.3 \pm 1.9}$	32.9 ± 0.4	$\mathbf{34.5 \pm 0.6}$
		HOSGNS$^{(stat\|dyn)}$	32.6 ± 0.5	36.2 ± 0.6	49.0 ± 1.8	32.9 ± 0.3	34.4 ± 0.5
	Time Embeddings	Positional Encoding	$\mathbf{73.1 \pm 1.1}$	$\mathbf{69.6 \pm 1.6}$	52.6 ± 1.9	$\mathbf{67.2 \pm 1.1}$	$\mathbf{62.1 \pm 1.4}$
		HOSGNS$^{(stat)}$	65.3 ± 1.2	57.3 ± 1.9	47.8 ± 1.3	57.3 ± 1.3	57.0 ± 1.0
		HOSGNS$^{(dyn)}$	72.2 ± 1.1	64.7 ± 1.8	52.1 ± 1.3	63.0 ± 1.3	60.5 ± 1.1
		HOSGNS$^{(stat\|dyn)}$	71.4 ± 0.7	64.7 ± 1.7	$\mathbf{53.0 \pm 1.3}$	61.2 ± 1.5	61.5 ± 1.1
$(0.125, 0.001)$	Node Embeddings	DEEPWALK	31.0 ± 0.4	38.5 ± 3.1	31.5 ± 1.0	32.1 ± 0.4	32.5 ± 0.4
		HOSGNS$^{(stat)}$	32.7 ± 0.5	42.6 ± 3.6	$\mathbf{43.8 \pm 2.5}$	$\mathbf{34.9 \pm 0.4}$	33.3 ± 0.4
		HOSGNS$^{(dyn)}$	32.2 ± 0.4	$\mathbf{44.5 \pm 3.4}$	42.5 ± 1.4	34.1 ± 0.4	$\mathbf{34.6 \pm 0.3}$
		HOSGNS$^{(stat\|dyn)}$	$\mathbf{33.4 \pm 0.4}$	43.5 ± 3.5	43.0 ± 2.4	33.7 ± 0.4	34.5 ± 0.4
	Time Embeddings	Positional Encoding	$\mathbf{72.2 \pm 0.8}$	$\mathbf{72.8 \pm 1.8}$	$\mathbf{60.3 \pm 1.9}$	$\mathbf{67.8 \pm 0.7}$	$\mathbf{61.3 \pm 0.9}$
		HOSGNS$^{(stat)}$	67.8 ± 1.7	63.0 ± 2.6	52.3 ± 2.1	59.4 ± 1.1	57.1 ± 0.9
		HOSGNS$^{(dyn)}$	70.5 ± 1.4	71.3 ± 2.0	57.5 ± 2.1	64.3 ± 1.0	60.0 ± 1.0
		HOSGNS$^{(stat\|dyn)}$	71.6 ± 1.1	70.9 ± 2.1	57.6 ± 1.8	65.4 ± 0.8	$\mathbf{61.3 \pm 0.9}$
$(0.0625, 0.002)$	Node Embeddings	DEEPWALK	31.7 ± 0.6	40.4 ± 3.6	32.1 ± 1.2	34.3 ± 0.4	33.4 ± 0.4
		HOSGNS$^{(stat)}$	33.4 ± 0.4	44.7 ± 3.4	$\mathbf{46.1 \pm 1.3}$	35.3 ± 0.3	33.7 ± 0.4
		HOSGNS$^{(dyn)}$	$\mathbf{33.7 \pm 0.4}$	$\mathbf{44.9 \pm 3.4}$	45.6 ± 1.5	$\mathbf{36.4 \pm 0.4}$	$\mathbf{36.1 \pm 0.4}$
		HOSGNS$^{(stat\|dyn)}$	33.2 ± 0.5	42.9 ± 2.9	43.0 ± 1.1	36.1 ± 0.5	35.7 ± 0.3
	Time Embeddings	Positional Encoding	$\mathbf{69.5 \pm 1.3}$	$\mathbf{67.0 \pm 2.4}$	$\mathbf{60.3 \pm 1.9}$	58.4 ± 0.6	54.6 ± 0.7
		HOSGNS$^{(stat)}$	64.0 ± 1.2	58.9 ± 2.7	47.0 ± 1.2	55.4 ± 0.7	54.7 ± 0.6
		HOSGNS$^{(dyn)}$	68.4 ± 0.7	66.3 ± 2.3	57.4 ± 1.6	58.8 ± 0.7	55.8 ± 0.6
		HOSGNS$^{(stat\|dyn)}$	67.7 ± 0.9	63.4 ± 2.5	54.9 ± 1.1	$\mathbf{59.3 \pm 0.6}$	$\mathbf{56.6 \pm 0.7}$
$(0.125, 0.002)$	Node Embeddings	DEEPWALK	31.0 ± 0.5	32.9 ± 0.5	31.8 ± 1.2	32.6 ± 0.4	33.2 ± 0.4
		HOSGNS$^{(stat)}$	$\mathbf{33.2 \pm 0.5}$	36.5 ± 0.4	43.6 ± 1.4	33.2 ± 0.5	33.5 ± 0.5
		HOSGNS$^{(dyn)}$	32.8 ± 0.5	37.8 ± 0.5	46.5 ± 1.5	34.4 ± 0.4	35.1 ± 0.4
		HOSGNS$^{(stat\|dyn)}$	33.0 ± 0.5	$\mathbf{38.1 \pm 0.7}$	$\mathbf{46.9 \pm 1.5}$	$\mathbf{34.6 \pm 0.5}$	$\mathbf{35.4 \pm 0.4}$
	Time Embeddings	Positional Encoding	$\mathbf{71.6 \pm 0.9}$	$\mathbf{66.6 \pm 1.5}$	$\mathbf{53.0 \pm 1.7}$	61.5 ± 0.8	56.8 ± 0.7
		HOSGNS$^{(stat)}$	63.3 ± 1.4	58.9 ± 1.2	45.2 ± 0.7	56.3 ± 0.9	56.0 ± 0.6
		HOSGNS$^{(dyn)}$	69.6 ± 0.8	62.8 ± 1.4	51.3 ± 1.4	$\mathbf{63.3 \pm 1.2}$	$\mathbf{58.6 \pm 0.7}$
		HOSGNS$^{(stat\|dyn)}$	71.0 ± 1.0	62.4 ± 1.1	52.7 ± 1.5	63.0 ± 0.8	58.2 ± 0.8
$(0.1875, 0.001)$	Node Embeddings	DEEPWALK	31.7 ± 0.5	35.7 ± 2.2	31.3 ± 1.0	31.8 ± 0.4	32.4 ± 0.3
		HOSGNS$^{(stat)}$	33.4 ± 0.5	38.1 ± 2.2	37.2 ± 1.2	33.4 ± 0.4	33.0 ± 0.4
		HOSGNS$^{(dyn)}$	32.4 ± 0.5	$\mathbf{38.3 \pm 2.2}$	38.0 ± 1.1	$\mathbf{34.3 \pm 0.4}$	$\mathbf{34.8 \pm 0.4}$
		HOSGNS$^{(stat\|dyn)}$	$\mathbf{34.0 \pm 0.5}$	38.2 ± 2.2	$\mathbf{39.7 \pm 1.4}$	33.8 ± 0.4	34.4 ± 0.4
	Time Embeddings	Positional Encoding	73.3 ± 0.6	$\mathbf{72.0 \pm 1.6}$	$\mathbf{66.9 \pm 1.9}$	$\mathbf{71.3 \pm 0.7}$	63.1 ± 1.1
		HOSGNS$^{(stat)}$	65.8 ± 1.0	60.9 ± 2.0	51.3 ± 1.7	59.9 ± 1.2	57.2 ± 0.9
		HOSGNS$^{(dyn)}$	71.7 ± 0.6	68.6 ± 1.7	63.3 ± 1.9	67.4 ± 0.9	$\mathbf{63.6 \pm 0.8}$
		HOSGNS$^{(stat\|dyn)}$	$\mathbf{73.5 \pm 0.6}$	69.4 ± 1.8	60.3 ± 2.2	66.8 ± 0.9	62.4 ± 0.9
$(0.125, 0.004)$	Node Embeddings	DEEPWALK	30.7 ± 0.3	32.0 ± 0.4	-	32.3 ± 0.4	-
		HOSGNS$^{(stat)}$	32.5 ± 0.4	35.3 ± 0.5		34.4 ± 0.4	
		HOSGNS$^{(dyn)}$	32.2 ± 0.4	$\mathbf{37.9 \pm 0.6}$		$\mathbf{34.6 \pm 0.3}$	
		HOSGNS$^{(stat\|dyn)}$	$\mathbf{32.7 \pm 0.5}$	36.9 ± 0.4		34.4 ± 0.4	
	Time Embeddings	Positional Encoding	68.4 ± 1.1	$\mathbf{61.3 \pm 1.2}$	-	58.2 ± 0.6	-
		HOSGNS$^{(stat)}$	62.6 ± 1.4	53.6 ± 1.4		53.5 ± 1.2	
		HOSGNS$^{(dyn)}$	67.1 ± 1.1	57.8 ± 0.9		58.3 ± 1.1	
		HOSGNS$^{(stat\|dyn)}$	$\mathbf{70.2 \pm 1.1}$	59.0 ± 1.1		$\mathbf{58.7 \pm 0.8}$	

Table 7: Macro-F1 scores deriving from the ablation study for temporal event reconstruction. Link and time representations, all with a dimension $d = 192$, are separately used as input to a Logistic Regression. We highlight in bold the best score, according to the type of predictor, for each dataset.

Predictors	Model			Dataset		
		LYONSCHOOL	SFHH	LH10	THIERS13	INVS15
Node Embeddings	DEEPWALK	82.9 ± 0.3	68.7 ± 0.5	53.8 ± 1.0	$\mathbf{90.2 \pm 0.2}$	79.1 ± 0.4
	HOSGNS$^{(stat)}$	83.5 ± 0.3	$\mathbf{72.8 \pm 0.4}$	$\mathbf{61.9 \pm 1.0}$	89.9 ± 0.2	$\mathbf{79.5 \pm 0.4}$
	HOSGNS$^{(dyn)}$	84.7 ± 0.3	68.4 ± 0.4	57.7 ± 1.0	90.1 ± 0.2	78.5 ± 0.5
	HOSGNS$^{(stat\|dyn)}$	$\mathbf{84.8 \pm 0.4}$	69.9 ± 0.5	58.1 ± 1.0	90.1 ± 0.2	77.9 ± 0.4
Time Embeddings	Positional Encoding	56.0 ± 0.4	51.9 ± 0.5	$\mathbf{51.3 \pm 0.7}$	$\mathbf{53.6 \pm 0.2}$	$\mathbf{53.1 \pm 0.4}$
	HOSGNS$^{(stat)}$	$\mathbf{57.0 \pm 0.4}$	50.5 ± 0.6	49.5 ± 0.5	52.7 ± 0.3	52.8 ± 0.3
	HOSGNS$^{(dyn)}$	55.3 ± 0.5	$\mathbf{52.6 \pm 0.6}$	50.5 ± 0.5	51.9 ± 0.3	52.9 ± 0.3
	HOSGNS$^{(stat\|dyn)}$	56.8 ± 0.4	51.4 ± 0.4	50.0 ± 0.6	51.6 ± 0.3	52.7 ± 0.3

A.10 EMPIRICAL COMPARISON WITH CLASSICAL TENSOR FACTORIZATION AND RELATIONAL LEARNING ALGORITHMS

Here we report in Tables 8 and 9 results about node classification of epidemic states and temporal event reconstruction comparing HOSGNS$^{(stat)}$ performances with HOLE (Nickel et al., 2015)[5], a 3-order relational learning algorithm, and HOSGNS$^{(dyn)}$ with 4-order CP decomposition (Kolda & Bader, 2009)[6]. Accounting LYONSCHOOL dataset, we trained HOLE with 30 epochs noticing a training time of 545s, and 10 iterations of tensor decomposition (4-CPTF) with a training time of 65305s. In displayed tables we show that HOSGNS performs better in both tasks compared with the corresponding baseline.

Table 8: Macro-F1 scores for classification of nodes in epidemic states according to different SIR epidemic processes for LYONSCHOOL dataset. Here HOSGNS variants are compared with tensor factorization and relational learning baselines, and tested with different operators to produce node-time representations used as input to a Logistic Regression. We highlight in bold the best scores between HOSGNS and baselines.

Operator	$(\beta,\mu)=(0.25,0.002)$ d=128		$(\beta,\mu)=(0.125,0.001)$ d=64	
	HOLE	HOSGNS$^{(stat)}$	4-CPTF	HOSGNS$^{(dyn)}$
Average	37.6 ± 1.4	$\mathbf{56.3 \pm 0.9}$	70.1 ± 1.1	$\mathbf{73.7 \pm 0.6}$
Hadamard	46.4 ± 1.0	$\mathbf{55.5 \pm 0.8}$	59.9 ± 0.9	$\mathbf{75.6 \pm 0.4}$
Weighted-L1	40.1 ± 1.5	$\mathbf{53.9 \pm 0.9}$	70.6 ± 1.1	$\mathbf{73.8 \pm 0.5}$
Weighted-L2	38.7 ± 1.1	$\mathbf{53.2 \pm 0.7}$	62.3 ± 1.6	$\mathbf{74.3 \pm 0.5}$
Concat	43.1 ± 1.8	$\mathbf{69.4 \pm 0.9}$	69.2 ± 0.6	$\mathbf{73.3 \pm 0.6}$

Table 9: Macro-F1 scores for temporal event reconstruction for LYONSCHOOL dataset. Here HOSGNS variants are compared with tensor factorization and relational learning baselines, and tested with different operators to produce link-time representations used as input to a Logistic Regression. We highlight in bold the best scores between HOSGNS and baselines.

Operator	d=128		d=64	
	HOLE	HOSGNS$^{(stat)}$	4-CPTF	HOSGNS$^{(dyn)}$
Average	49.6 ± 0.4	$\mathbf{61.2 \pm 0.4}$	57.8 ± 0.6	$\mathbf{67.1 \pm 0.4}$
Hadamard	70.9 ± 0.4	$\mathbf{98.5 \pm 0.1}$	71.3 ± 0.6	$\mathbf{89.6 \pm 0.2}$
Weighted-L1	54.2 ± 0.5	$\mathbf{67.2 \pm 0.4}$	56.5 ± 0.6	$\mathbf{79.4 \pm 0.3}$
Weighted-L2	50.1 ± 0.6	$\mathbf{68.5 \pm 0.3}$	56.7 ± 0.4	$\mathbf{78.9 \pm 0.5}$
Concat	62.9 ± 0.4	$\mathbf{63.3 \pm 0.5}$	63.1 ± 0.6	$\mathbf{64.7 \pm 0.5}$

A.11 MODEL PERFORMANCE ON SYNTHETIC DATA

Here we report in Tables 10 and 11 results about node classification of epidemic states and temporal event reconstruction on synthetic datasets. Contrary to what previously declared, in this section HOSGNS is trained directly sampling random walks from $\{\mathcal{G}^{(k)}\}_{k\in\mathcal{T}}$ and $\mathcal{G}_{\mathcal{H}}$ for HOSGNS$^{(stat)}$ and HOSGNS$^{(dyn)}$ respectively with window sizes $T=1,10$. The batch size of positive examples is fixed to 20000, and for each element in the batch $\kappa=5$ negative tuples are sampled from the corpus. Embedding parameters are initialized with 1000 warm-up steps. These results are line with the ones previously shown in the main paper on empirical datasets.

---

[5] `https://github.com/mnick/scikit-kge`

[6] `https://github.com/tensorly/tensorly`

Table 10: Macro-F1 scores for classification of nodes in epidemic states according to different SIR epidemic processes for synthetic datasets. Here for each HOSGNS variant we tested different operators to produce node-time representations, all with a dimension $d = 128$, used as input to a Logistic Regression. For each (β, μ) we highlight the two highest scores and underline the best one.

(β, μ)	Model	Operator	Dataset	
			OPENABM-COVID19-2k-100	OPENABM-COVID19-5k-20
$(0.25, 0.002)$	DYANE	-	57.9 ± 1.8	59.6 ± 1.7
	HOSGNS$^{(stat)}$	Average	31.2 ± 0.1	27.9 ± 0.5
		Hadamard	31.2 ± 0.1	27.8 ± 0.6
		Weighted-L1	31.1 ± 0.1	28.1 ± 0.9
		Weighted-L2	31.3 ± 0.2	27.6 ± 0.6
		Concat	32.4 ± 1.1	27.8 ± 0.9
	HOSGNS$^{(dyn)}$	Average	$\underline{\mathbf{61.3}} \pm \mathbf{1.3}$	$\mathbf{60.6} \pm \mathbf{1.3}$
		Hadamard	57.5 ± 1.8	$\underline{\mathbf{61.0}} \pm \mathbf{1.1}$
		Weighted-L1	56.5 ± 1.8	56.5 ± 1.9
		Weighted-L2	$\mathbf{60.3} \pm \mathbf{1.3}$	55.8 ± 2.3
		Concat	49.2 ± 2.0	56.7 ± 1.8
$(0.125, 0.001)$	DYANE	-	$\mathbf{61.6} \pm \mathbf{1.2}$	$\underline{\mathbf{60.6}} \pm \mathbf{0.7}$
	HOSGNS$^{(stat)}$	Average	31.5 ± 0.2	24.6 ± 1.3
		Hadamard	31.5 ± 0.2	24.8 ± 1.3
		Weighted-L1	31.5 ± 0.2	25.1 ± 1.1
		Weighted-L2	31.4 ± 0.2	23.8 ± 1.3
		Concat	30.9 ± 1.0	27.6 ± 1.7
	HOSGNS$^{(dyn)}$	Average	60.3 ± 1.5	60.3 ± 0.8
		Hadamard	61.3 ± 1.0	60.1 ± 1.1
		Weighted-L1	$\underline{\mathbf{62.9}} \pm \mathbf{0.3}$	55.1 ± 2.3
		Weighted-L2	60.0 ± 1.4	55.3 ± 2.2
		Concat	60.0 ± 1.1	$\mathbf{60.4} \pm \mathbf{1.1}$
$(0.0625, 0.002)$	DYANE	-	$\underline{\mathbf{61.8}} \pm \mathbf{0.4}$	53.8 ± 1.3
	HOSGNS$^{(stat)}$	Average	29.9 ± 0.2	30.1 ± 1.4
		Hadamard	29.8 ± 0.2	29.4 ± 1.4
		Weighted-L1	29.8 ± 0.3	29.6 ± 0.9
		Weighted-L2	30.0 ± 0.2	30.3 ± 1.1
		Concat	30.8 ± 1.0	32.1 ± 1.9
	HOSGNS$^{(dyn)}$	Average	$\mathbf{61.4} \pm \mathbf{0.5}$	$\underline{\mathbf{57.4}} \pm \mathbf{1.9}$
		Hadamard	59.5 ± 0.9	$\mathbf{54.5} \pm \mathbf{1.4}$
		Weighted-L1	60.2 ± 1.0	51.5 ± 2.4
		Weighted-L2	61.3 ± 0.3	46.4 ± 1.9
		Concat	60.7 ± 0.5	54.3 ± 2.1
$(0.125, 0.002)$	DYANE	-	60.7 ± 1.1	$\mathbf{61.3} \pm \mathbf{0.6}$
	HOSGNS$^{(stat)}$	Average	30.8 ± 0.2	26.6 ± 1.2
		Hadamard	30.8 ± 0.1	27.4 ± 1.2
		Weighted-L1	30.5 ± 0.2	25.1 ± 1.3
		Weighted-L2	31.0 ± 0.2	24.6 ± 1.2
		Concat	31.3 ± 1.1	27.1 ± 1.8
	HOSGNS$^{(dyn)}$	Average	61.3 ± 0.9	$\underline{\mathbf{61.7}} \pm \mathbf{0.7}$
		Hadamard	58.9 ± 1.4	60.7 ± 0.6
		Weighted-L1	$\underline{\mathbf{62.1}} \pm \mathbf{0.5}$	56.2 ± 2.4
		Weighted-L2	$\mathbf{61.7} \pm \mathbf{0.5}$	54.3 ± 2.1
		Concat	58.7 ± 0.9	59.4 ± 1.3
$(0.1875, 0.001)$	DYANE	-	$\mathbf{60.3} \pm \mathbf{1.4}$	59.6 ± 1.5
	HOSGNS$^{(stat)}$	Average	32.0 ± 0.2	25.2 ± 1.0
		Hadamard	31.9 ± 0.2	27.4 ± 0.7
		Weighted-L1	31.9 ± 0.2	26.6 ± 0.8
		Weighted-L2	31.9 ± 0.2	26.0 ± 0.7
		Concat	30.2 ± 0.4	30.8 ± 1.4
	HOSGNS$^{(dyn)}$	Average	58.8 ± 1.6	$\underline{\mathbf{61.6}} \pm \mathbf{1.2}$
		Hadamard	$\underline{\mathbf{60.5}} \pm \mathbf{1.1}$	$\mathbf{60.9} \pm \mathbf{1.0}$
		Weighted-L1	59.5 ± 1.7	57.4 ± 1.9
		Weighted-L2	59.3 ± 1.7	56.5 ± 2.2
		Concat	54.5 ± 1.8	59.9 ± 1.0
$(0.125, 0.004)$	DYANE	-	$\mathbf{60.0} \pm \mathbf{1.1}$	$\underline{\mathbf{60.8}} \pm \mathbf{0.6}$
	HOSGNS$^{(stat)}$	Average	29.7 ± 0.2	25.8 ± 1.1
		Hadamard	29.5 ± 0.2	27.0 ± 1.2
		Weighted-L1	29.6 ± 0.2	26.0 ± 0.8
		Weighted-L2	29.9 ± 0.2	23.7 ± 1.2
		Concat	30.8 ± 0.7	30.0 ± 1.6
	HOSGNS$^{(dyn)}$	Average	58.6 ± 1.6	59.5 ± 1.1
		Hadamard	58.4 ± 1.2	$\mathbf{60.1} \pm \mathbf{0.7}$
		Weighted-L1	$\underline{\mathbf{60.8}} \pm \mathbf{1.0}$	56.2 ± 1.9
		Weighted-L2	59.2 ± 1.2	52.1 ± 2.6
		Concat	58.0 ± 1.3	59.2 ± 1.1

Table 11: Macro-F1 scores for temporal event reconstruction for synthetic datasets. Here for each HOSGNS variant we tested different operators to produce link-time representations, all with a dimension $d = 192$, used as input to a Logistic Regression. We highlight in bold the best two overall scores for each dataset. For baseline models we underline their highest score.

Model	Operator	Dataset	
		OPENABM-COVID19-2k-100	OPENABM-COVID19-5k-20
DYANE	Average	52.2 ± 0.1	51.9 ± 0.1
	Hadamard	$\underline{76.4} \pm 0.1$	$\mathbf{\underline{90.5} \pm 0.3}$
	Weighted-L1	70.3 ± 0.1	78.2 ± 0.7
	Weighted-L2	70.3 ± 0.1	78.8 ± 0.5
	Concat	53.8 ± 0.1	52.5 ± 0.1
HOSGNS$^{(stat)}$	Average	54.6 ± 0.1	55.1 ± 0.2
	Hadamard	$\mathbf{91.1 \pm 0.1}$	$\mathbf{98.7 \pm 0.1}$
	Weighted-L1	69.8 ± 0.1	72.7 ± 0.1
	Weighted-L2	72.7 ± 0.1	76.6 ± 0.1
	Concat	56.5 ± 0.1	57.4 ± 0.1
HOSGNS$^{(dyn)}$	Average	54.0 ± 0.2	54.7 ± 0.1
	Hadamard	$\mathbf{78.7 \pm 0.1}$	82.8 ± 0.3
	Weighted-L1	71.5 ± 0.3	78.5 ± 0.1
	Weighted-L2	73.1 ± 0.2	80.5 ± 0.1
	Concat	57.1 ± 0.1	57.5 ± 0.1

REFERENCES

Alain Barrat and Ciro Cattuto. Temporal networks of face-to-face human interactions. In *Temporal Networks*, pp. 191–216. Springer, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Edoardo Galimberti, Alain Barrat, Francesco Bonchi, Ciro Cattuto, and Francesco Gullo. Mining (maximal) span-cores from temporal networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 107–116, 2018.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

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

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3): 455–500, 2009.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pp. 2177–2185, 2014.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. *arXiv preprint arXiv:1510.04935*, 2015.

André Panisson, Laetitia Gauvin, Alain Barrat, and Ciro Cattuto. Fingerprinting temporal networks of close-range human proximity. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 261–266. IEEE, 2013.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proc. of the 20th ACM SIGKDD Int.l Conf. on Knowledge Discovery and Data Mining*, pp. 701–710. ACM, 2014.

Anna Sapienza, Alain Barrat, Ciro Cattuto, and Laetitia Gauvin. Estimating the outcome of spreading processes on networks with incomplete information: A dimensionality reduction approach. *Physical Review E*, 98(1):012317, 2018.

Michele Starnini, Andrea Baronchelli, Alain Barrat, and Romualdo Pastor-Satorras. Random walks on temporal networks. *Physical Review E*, 85(5):056115, 2012.

Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one*, 6(8), 2011.

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*, pp. 5998–6008, 2017.