Temporal Graph AutoEncoder: Mapping Dynamic Graphs to Dynamical Systems with Neural ODEs

Raphaël Romero
Ghent University
raphael.romero@ugent.be

Rembert Daems
Ghent University – IMEC
rembert.daems@ugent.be

Tijl De Bie
Ghent University
tijl.debie@ugent.be

Abstract

Representation learning on dynamic graphs has gained increasing attention due to its wide-ranging applications. A common approach is to update node embeddings incrementally, where each new event modifies a stored memory state in a jump-driven manner. For link prediction, however, such methods often end up modeling only short-term correlations between the most recent events and the next ones, rather than the underlying dynamics of the network as an interacting system. Moreover, it remains unclear to what extent a dynamic network can be effectively modeled using *history-free* dynamic embeddings. This work addresses this question by proposing a novel TGAE (Temporal Graph AutoEncoder). TGAE leverages Neural ODEs to represent nodes with time-varying embeddings defined by an initial condition and a global vector field parameterized by a neural network. Pairwise interactions are modeled as an inhomogeneous Poisson process, with rates determined by latent-space distances, enabling self-supervised training from observed events. Experiments on synthetic data and a high school contact-tracing dataset demonstrate that TGAE captures temporal patterns and extrapolates beyond the training horizon.

1 Introduction

Dynamic graph representation learning seeks to capture the temporal evolution of a network by modeling node trajectories that evolve in response to observed interactions (u, v, t) between them. A key challenge is to design models that not only summarize past dynamics but also generalize to predict future, unseen interactions. Existing approaches fall broadly into two categories. **Time-transductive latent space models**, such as the Continuous Latent Position Model (CLPM) [1] and Intensity Profile Projection [2], directly fit trajectories of node embeddings to the observed event history. These models yield history-independent summaries of past interactions, but their transductive nature prevents extrapolation beyond the training window. In contrast, **Time-inductive event-conditioned models**, such as Jodie [3], DyRep [4], TGN [5], and TGAT [6], generate embeddings for future timestamps by conditioning on past events. Here, the underlying dynamics are inherently history-dependent: embeddings evolve only when new interactions occur. This perspective can be formalized as a jump-controlled ODE [7], where jumps are triggered by the edge-level event process $\boldsymbol{Y}_{ij}(t)$:

$$\mathrm{d}z(t) = f_\theta(z(t), t)\,\mathrm{d}t + \sum_{i,j} h_\theta\big(z_i(t), z_j(t)\big)\,\mathrm{d}\boldsymbol{Y}_{ij}(t),$$

where $z(t)$ denotes the collection of node embeddings. The flow term f_θ governs continuous evolution, while h_θ models discrete event-driven jumps. As noted in [8], most existing methods emphasize jump dynamics as a natural extension of message-passing GNNs, often neglecting the flow term. Moreover, training is typically performed on node-level supervised tasks. By contrast, the potential of learning

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Advancing Graph Machine Learning.

a history-independent flow term as a basis for *self-supervised representation learning* on dynamic graphs remains largely unexplored. In this paper, we take a step toward closing this gap by asking:

Q1: can one learn a history-independent latent dynamical system that nonetheless extrapolates to future interactions?.

To address this question, we propose **T**emporal **G**raph **A**uto**E**ncoder (TGAE), a simple model where all node embeddings evolve jointly under a neural global vector field, trained on past interactions yet defining an autonomous system that extrapolates to future times.

Our main **contributions** can be summarized as follows:

1. We address question Q1 by studying a simple neural ODE model in which all node embeddings evolve jointly under a *global vector field*. The model is trained only on past interactions but defines an autonomous dynamical system that can be extrapolated to future times. This provides a compact, history-free summary of the graph's dynamics that fills the gap between static embeddings and fully event-conditioned architectures.
2. We show that this model is able to reproduce simple but common macroscopic temporal patterns, such as time-varying interaction rates and periodic community structure.

2 Background and Notations

Let $\mathcal{T} = [0, T]$ denote the observation interval, and let $\mathcal{U}$ be a fixed set of nodes known in advance. **Dynamic Graph Data** consists of a sequence of interactions $\{(u_m, v_m, t_m)\}_{m=1}^{M}$ with $0 < t_1 < \ldots < t_M < T$, where each triplet (u_m, v_m, t_m) represents an interaction between nodes $u_m, v_m \in \mathcal{U}$ at time t_m. This sequence can be represented compactly by a matrix of counting measures $\boldsymbol{Y}$, such that for any interval $[a, b] \subset \mathcal{T}$, $\boldsymbol{Y}([a, b])$ records the number of interactions between each node pair occurring within $[a, b]$.

A Poisson Process Model We consider the interactions as arising from an Inhomogeneous Poisson Process. This means that for any interval $[a, b] \subset \mathcal{T}$, the count of interactions between node pairs is distributed according to $\boldsymbol{Y}([a, b]) \sim \text{Poisson}(\Lambda([a, b]))$. We denote this using the shorthand notation $\boldsymbol{Y} \sim \text{PoissonProcess}(\Lambda)$.

Neural ODEs Neural Ordinary Differential Equations (NODEs) [9] provide a framework for learning dynamical systems formulated as an initial value problem $(f_\theta, z(0))$, where f_θ is a neural network with parameters θ. The latent state is then available in continuous time by integration via an ODE solver:

$$z(t) = z(0) + \int_0^t f_\theta(z(s))\ \mathrm{d}s.$$

The tools from [9] and related work enable efficient backpropagation through the ODE solver, allowing learning both the initial condition $z(0)$ and the dynamics f_θ.

3 Temporal Graph AutoEncoder (TGAE)

In Temporal Graph AutoEncoder (TGAE), we encode the dynamic graph into a global continuous-time latent state $z(t) = [z_1(t), \ldots, z_n(t)]^T \in \mathbb{R}^{n \times d}$, where $z_i(t)$ is the embedding of node i at time t. Its evolution follows the Neural ODE $\frac{\mathrm{d}z(t)}{\mathrm{d}t} = f_\theta(z(t), t)$, parameterized by neural network weights θ and initial condition $z(0)$, which are both learned from data.

Form of the Dynamic Function f_θ. The system dynamics map the current state and time information to a direction of evolution in the joint embedding space. We require smooth temporal evolution and explicit dependence on time, introduced via Fourier time-encoding features. The dynamic function takes the form:

$$z(t) \mapsto f_\theta(z(t), t) = W_{\text{out}}.\big(\alpha_\theta(\Phi(t)) \odot h_\theta(z(t))\big)$$

where h_θ is a state-dependent hidden representation, $\alpha_\theta(\Phi(t))$ is a learned scale vector obtained from a time-embedding $\Phi(t)$, and $\odot$ denotes elementwise multiplication. The time embedding is defined as

$$\Phi(t) = \big[\cos(2\pi k t/T),\ \sin(2\pi k t/T)\big]_{k=1}^{K},$$

so that the dynamics are explicitly periodic in t with resolution controlled by K (number of Fourier modes) and T (base period).

Decoder (Generative Model) The decoder assigns a Poisson rate to each node pair at any time t, yielding a fully inductive generative model for dynamic graphs. This enables simulation via the thinning algorithm (see Alg. 1). Inspired by Euclidean distance models [10, 1] we define the rate function as:

$$\mathbf{\Lambda}_{u,v}(t) = \exp\left(\beta - ||z_u(t) - z_v(t)||^2\right), \tag{1}$$

where the intercept term β can be either learned jointly with the model parameters or fixed as a hyperparameter.

Training We present two related training strategies. *(1) Maximum likelihood:* the model is treated as a point process with dynamics

$$\begin{cases} \dfrac{\mathrm{d}z}{\mathrm{d}t} = f_{\underline{\theta}}(z,t),\ z(0) = \underline{z_0}, \\ \mathbf{\Lambda}_{u,v}(t) = \exp\big(\beta - ||z_u(t) - z_v(t)||^2\big), \\ \boldsymbol{Y}_{u,v} \sim \mathrm{PoissonProcess}(\mathbf{\Lambda}_{u,v}) \end{cases}$$

which leads to the loss function

$$\mathcal{L}(z(0),\theta) = \sum_{u \neq v} \left[\int_0^T \lambda_{u,v}(t)\,\mathrm{d}t - \int_0^T \lambda_{u,v}(t)\,\mathrm{d}\boldsymbol{Y}_{u,v}(t) \right] \tag{2}$$

and parameters are estimated by minimizing the negative log-likelihood of the observed events. *(2) Binary classification:* for each observed event (u, v, t) we compute a score $s_{\text{pos}}(u, v, t) = \beta - \|z_u(t) - z_v(t)\|^2$ and generate negative samples (u, v', t) at each epoch, then minimize the binary cross-entropy loss to discriminate positive from negative events. This strategy avoids the burden of calculating the compensator (continuous integral) term for all the node pairs.

Algorithm 1 Thinning Method for Jointly Integrating ODE and Sampling Events

Input: Initial condition z_0, initial time $t0$, final time T, fitted ODE model $z, t \mapsto f(z(t), t, \theta)$
Initialize:
$\mathcal{H} \leftarrow \emptyset$
$t \leftarrow t0$
while $t < T$ **do**
  Calculate the Poisson rates $\lambda_{ij}(t)$ using $\lambda_{ij}(t) = \exp(\beta - \|z_i(t) - z_j(t)\|^2)$.
  Calculate the cumulative rate $\Lambda(t) = \sum_{i,j} \lambda_{ij}(t)$.
  Sample a step size Δt from an exponential distribution: $\Delta t \sim \mathrm{Exp}(\Lambda(t))$
  Sample a node-pair (i, j) from the set of all possible node pairs using a Multinomial distribution with probabilities $p(i, j, t) = \frac{\lambda_{ij}(t)}{\Lambda(t)}$.
  Add the event (i, j, t) to the event list $\mathcal{H}$.
  Solve the ODE system $z(t)$ from t to $t + \Delta t$ using the chosen solver (for instance Euler's Method):

$$z(t + \Delta t) \approx z(t) + \int_t^{t+\Delta t} f(z(s), s, \theta)\,\mathrm{d}s.$$

  Set $t \leftarrow t + \Delta t$
end while
return ODE solution $z(t)$ and event list $\mathcal{E}$

Modeling Stochasticity. While macroscopic patterns may emerge in a dynamic network, individual interactions are intrinsically stochastic and may also reflect unobserved factors. This variability can be captured by extending the deterministic ODE formulation with a diffusion term, leading to a stochastic differential equation (SDE)[11, 12, 13]. Denoting by $W(t)$ a Wiener process and by $g_\theta(z(t), t)$ a diffusion neural network that parameterizes the noise intensity, the dynamics become:

$$\mathrm{d}z(t) \;=\; f_\theta(z(t), t)\,\mathrm{d}t \;+\; g_\theta(z(t), t)\,\mathrm{d}W(t).$$

This formulation preserves the learned structure of the drift f_θ while explicitly modeling uncertainty and random fluctuations in the trajectories.

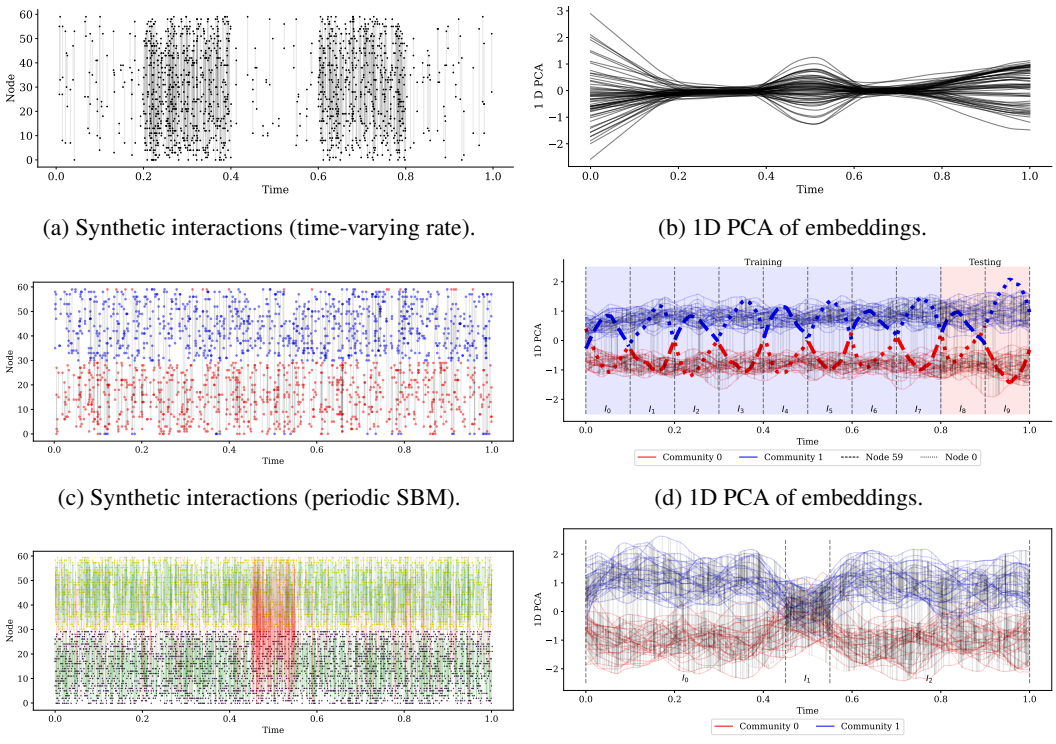
4 Experiments

We empirically evaluate TGAE on three synthetic scenarios: (a) time-varying interaction rates, (b) periodic community switching, and (c) communities merging over time. Full details on the data generating process are provided in Appendix B. All experiments were run on a Macbook Air M1 with 8 GB RAM. The code is available at `https://github.com/aida-ugent/tgae`.

Time-varying interaction rates We simulate node interactions with rates alternating between high (0.10) and low (0.01) over five intervals within $[0, 1]$, testing whether TGAE can capture non-stationary patterns (see details in Appendix B.1). As shown in Figure 1b, the embeddings contract during high-rate intervals and expand during low-rate intervals, illustrating the model's ability to capture temporal fluctuations in overall interaction intensity.

Nodes switching communities over time We generate data from a periodic stochastic block model where nodes 0 and 59 swap communities every 0.1 time units, while other nodes maintain fixed community assignments (see Appendix B.2). This evaluates the model's ability to track dynamic community structure. We assess representation quality using clustering metrics (Adjusted Rand Index) applied to learned embeddings. The results are shown in Figure 1d. In the model fit shown on the Figure, the Train ARI reached 0.997, and Test ARI: 0.993, showing that the model accurately captured the periodic community structure.

Communities Merging over time We also test TGAE on a scenario where two communities merge temporarily before splitting again. Inter- and intra-community rates are set to 0.25 and 4.75 respectively on intervals $[0, 0.45]$ and $[0.55, 1]$, and both equal 2.5 on $[0.45, 0.55]$ (community structure disappears)(see Appendix B.3). Results in Figure 1f show the model successfully captures this dynamic, with embeddings merging during the middle interval and splitting again afterwards.



(a) Synthetic interactions (time-varying rate).

(b) 1D PCA of embeddings.

(c) Synthetic interactions (periodic SBM).

(d) 1D PCA of embeddings.

(e) Synthetic interactions (merging communities).

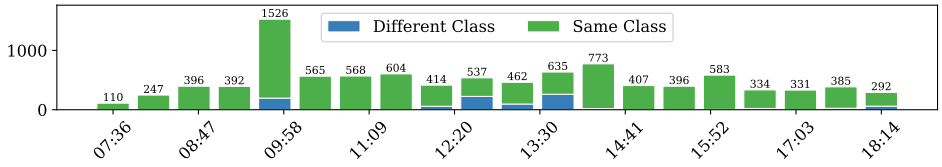
(f) 1D PCA of embeddings.

Figure 1: Synthetic data experiments. (Top) Time-varying interaction rate ($d = 8$, $K = 16$, $T = 1$). (Middle) Nodes periodically switching communities ($d = 8$, $K = 16$, $T = 0.1$). (Bottom) Two communities merging over time ($d = 8$, PCA reduced to 1D).

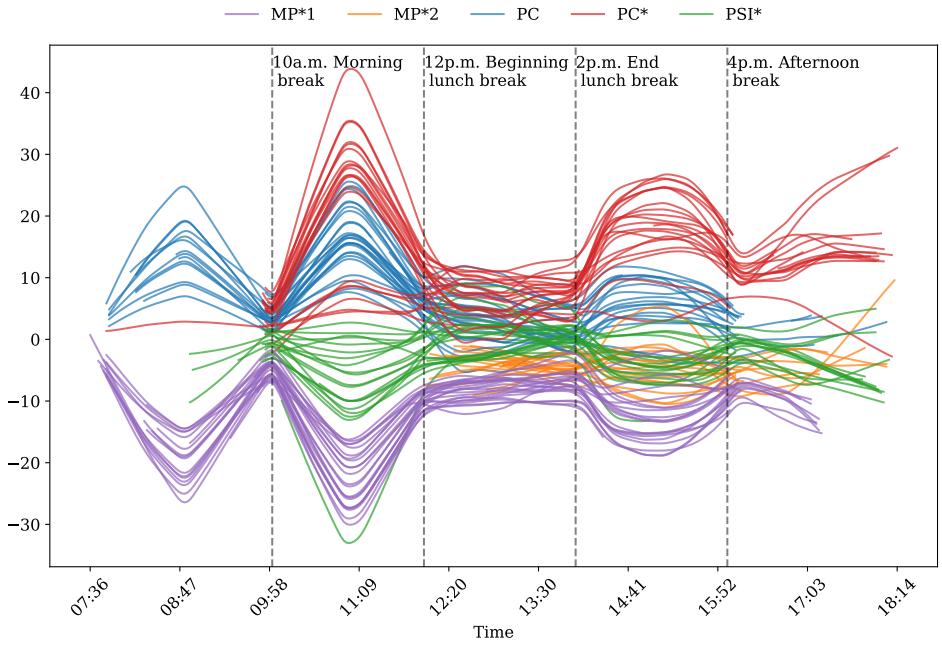
4.1 Embeddings visualization on the High-School Data

Evaluation Setting We consider a dataset of face-to-face interaction between students in a French high school, collected by [14]. The dataset consists of a sequence of interactions between students, recorded at a resolution of 20 seconds. In this paper we study the first day of the dataset, in which 180 students have a total of 9957 interactions between 6:30 a.ms and 5:30 p.m. The students are divided into 5 classes. The dominant pattern in the data is that inter-class interactions are much more frequent during break and lunch times, while intra-class interactions dominate during class time. The dataset was downloaded from `https://www.sociopatterns.org/datasets/high-school-contact-and-friendship-networks/` and made available in [14].

Visualization We train the TGAE model on one day of interactions, using 1-dimensional embeddings, and a time embedding of dimension 16. The model is trained using the binary cross-entropy loss. The trajectory of each student is only displayed between the time of their first and last interaction times. The resulting trajectories are shown in Figure 2b. This visualization shows the flow of the latent state over time, and exhibits a clear pattern of breaks and lunch times. In general, the model learns to separate the students into their respective classes. Moreover, during lunch break (around 12:30), more crossings can be observed. The model learns trajectories such that the nodes intersect at those times. This is consistent with the fact that during lunch break, students tend to interact more, and more frequently with students from other classes, as shown in Figure 2a.



(a) Count of events over time, split into inter-class and intra-class interactions. Breaks and lunch time typically see a more inter-class interactions.



(b) 1-dimensional trajectories of the students, colored by class. The trajectories are obtained by fitting the TGAE model with a dimension of 1.

5 Related Work

Dynamic Graph Neural Networks Methods like JODIE [3], DyRep [4], and TGN [5] extend static graph neural networks to Dynamic graphs by learning a complex function which updates embeddings based on recent events. This enables predictions but makes the embeddings dependent on event history. TGAE, in contrast, learns a data-independent dynamic, offering a distinct approach to temporal graph representation learning that is not tied to specific past interactions.

Neural ODEs for Dynamic Graphs Several studies have investigated the application of Neural Ordinary Differential Equations (Neural ODEs) to graph-structured data. GRAND [15] and CGNN [16] leverage Neural ODEs to define continuous-depth graph neural networks. Han et al. [17], on the other hand, extend this idea to link prediction in temporal knowledge graphs. Closer to our work, Poli et al. [7] introduce Neural Graph Differential Equations (Neural GDEs), which integrate Neural ODEs with graph neural networks for dynamic graph prediction. However, these approaches still rely on a discrete sequence of adjacency matrices rather than directly modeling the *continuous-time* edge-level point processes underlying graph evolution. Furthermore, they typically employ history-dependent, jump-based dynamics, whereas our method learns a deterministic, history-independent flow function that governs the temporal evolution of the graph.

Point Process Models Point process models provide a principled way to model event sequences in continuous time. The Neural Hawkes Process [18] uses an RNN to modulate the intensity function. Recent work has combined point processes with graph neural networks - for example, [19] propose a geometric Hawkes process using graph convolutions. Our work differs by learning a latent dynamic independent of the event history.

Latent Space Models Latent space models for networks [10] embed nodes in a latent space such that their proximity determines edge probabilities. CLPM [1] extends this to temporal networks by learning smooth latent trajectories. However, unlike our approach, CLPM is time-transductive. Our work bridges the gap between time-inductive methods and latent space models by learning data-independent dynamics.

6 Conclusions

Summary In this paper, we present TGAE, a new model for learning time-varying node representations in continuous-time dynamic graphs. The key innovation lies in using neural ODEs to embed dynamic graphs into a deterministic, history-independent dynamical system. Our experiments on synthetic data show that this approach can capture temporal patterns such as time-varying interaction rates and community structures. We further validate these capabilities on real data from a high school contact network, where TGAE learns meaningful trajectories that reflect temporal patterns such as breaks and lunch periods. The model successfully separates students by class while capturing cross-class interactions during social periods.

Limitations Our approach presents two limitations. First, the assumption of smooth dynamic may not hold in all scenarios, where in dynamic graphs, abrupt changes can occur due to sudden events. Secondly, the model parameters stores linearly with the number of nodes. This currently limits scalability to very large graphs but can be easily adressed in presence of initial node features, by using a neural network to parameterize the initial condition. Finallly, real-world datasets preent a combination of macroscopic, learnable patterns, and microscopic, unpredictable events. Our current approach focuses on learning the former using a deterministic ODE, but cannot capture the latter. Understanding how much of the dynamics can be learned, and how to model the remaining uncertainty, remains an open question.

Future Work Several promising directions remain for future work. On the one hand, understanding the theoretical effect of adding jumps to the ODE dynamics, (e.g. via Graph Neural Networks as in [7]), seems like a promising direction. As outlined in section , studying the capacity of this approach to capture uncertainty with stochastic differential equations, and combining it with variational inference via methods such as [12] could lead to a Temporal Graph Variational Autoencoder. In summary, the proposed TGAE model opens up new possibilities for the development and theoretical understanding of dynamic graph representation learning methods.

Acknowledgments and Disclosure of Funding

The research leading to these results has received funding from the Special Research Fund (BOF) of Ghent University (BOF20/IBF/117), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, from the FWO (project no. G0F9816N, 3G042220, G073924N). Funded by the European Union (ERC, VIGILIA, 101142229). Furthermore, it was supported by Flanders Make under the SBO project CADAIVISION. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Riccardo Rastelli and Marco Corneli. Continuous latent position models for instantaneous interactions. *Network Science*, pages 1–29, July 2023.

[2] Alexander Modell, Ian Gallagher, Emma Ceccherini, Nick Whiteley, and Patrick Rubin-Delanchy. Intensity Profile Projection: A Framework for Continuous-Time Representation Learning for Dynamic Networks, January 2024.

[3] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1269–1278, Anchorage AK USA, July 2019. ACM.

[4] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. DYREP: LEARNING REPRESENTATIONS OVER DYNAMIC GRAPHS. 2019.

[5] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal Graph Networks for Deep Learning on Dynamic Graphs. *arXiv*, pages 1–16, 2020.

[6] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive Representation Learning on Temporal Graphs. *arXiv*, pages 1–19, 2020.

[7] Michael Poli, Stefano Massaroli, Clayton M. Rabideau, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Continuous-Depth Neural Models for Dynamic Graph Prediction, June 2021.

[8] Alessio Gravina, Daniele Zambon, Davide Bacciu, and Cesare Alippi. Temporal graph odes for irregularly-sampled time series. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024.

[9] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[10] Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[11] Rembert Daems, Manfred Opper, Guillaume Crevecoeur, and Tolga Birdal. Variational Inference for SDEs Driven by Fractional Noise, October 2023.

[12] Manfred Opper. Variational Inference for Stochastic Differential Equations. *Annalen der Physik*, 531(3):1800233, 2019.

[13] Patrick Kidger, James Foster, Xuechen Li, Harald Oberhauser, and Terry Lyons. Neural SDEs as Infinite-Dimensional GANs.

[14] Julie Fournet and Alain Barrat. Contact Patterns among High School Students. *PLOS ONE*, 9(9):e107878, September 2014.

[15] Ben Chamberlain, James Rowbottom, Maria I. Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph Neural Diffusion. In *Proceedings of the 38th International Conference on Machine Learning*, pages 1407–1418. PMLR, July 2021.

[16] Louis-Pascal A. C. Xhonneux, Meng Qu, and Jian Tang. Continuous Graph Neural Networks, July 2020.

[17] Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. Learning Neural Ordinary Equations for Forecasting Future Links on Temporal Knowledge Graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

[18] Hongyuan Mei and Jason M Eisner. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process.

[19] Jin Shang and Mingxuan Sun. Geometric Hawkes Processes with Graph Convolutional Recurrent Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4878–4885, July 2019.

[20] Patrick Kidger. *On Neural Differential Equations*. PhD thesis, University of Oxford, 2021.

A Implementation

Our implementation is based on the library diffrax [20] for solving Neural ODEs. Upon acceptance, we will release an implementation of the method in Jax.

B Synthetic Dataset

B.1 Time-Varying interaction rates

We construct a synthetic dataset to test whether TGAE can capture time-varying interaction rates. The time horizon is $T = 1$ with $n = 30$ nodes. We partition $[0, 1]$ into five intervals

$$I_1 = [0, 0.2), \quad I_2 = [0.2, 0.4), \quad I_3 = [0.4, 0.6), \quad I_4 = [0.6, 0.8), \quad I_5 = [0.8, 1].$$

The interaction rate alternates between low and high values:

$$\lambda_{ij}(t) = \begin{cases} 0.01, & t \in I_1 \cup I_3 \cup I_5, \\ 0.10, & t \in I_2 \cup I_4. \end{cases}$$

To do so, for each unordered pair (i, j) and interval I_k, we sample

$$N_{ij}^{(k)} \sim \text{Poisson}(\lambda_{ij}(t)\,|I_k|)\,,$$

and place the $N_{ij}^{(k)}$ events uniformly at random within I_k.

B.2 Nodes Changing Community Over Time

Data Generation We generate data from a periodic stochastic block model where community memberships can vary with time. Specifically, nodes 0 and 59 alternate communities every 0.1 time units: starting with $c_0(0) = 0$ and $c_{59}(0) = 1$, they swap at each 0.1 interval. All other nodes have fixed communities: nodes 1–30 belong to community 0, and nodes 31–58 to community 1. Thus, within each interval of length 0.1, the community structure remains constant.

For a given interval with assignment $\{c_i\}$, interactions between node pairs (i, j) are sampled as

$$n_{ij} \sim \text{Poisson}(\lambda_{ij}), \quad \text{where} \quad \lambda_{ij} = \begin{cases} 0.5, & c_i = c_j, \\ 0.0125, & c_i \neq c_j, \end{cases}$$

and interaction times are drawn uniformly within the interval. We repeat this over multiple intervals and concatenate the results to obtain the dataset.

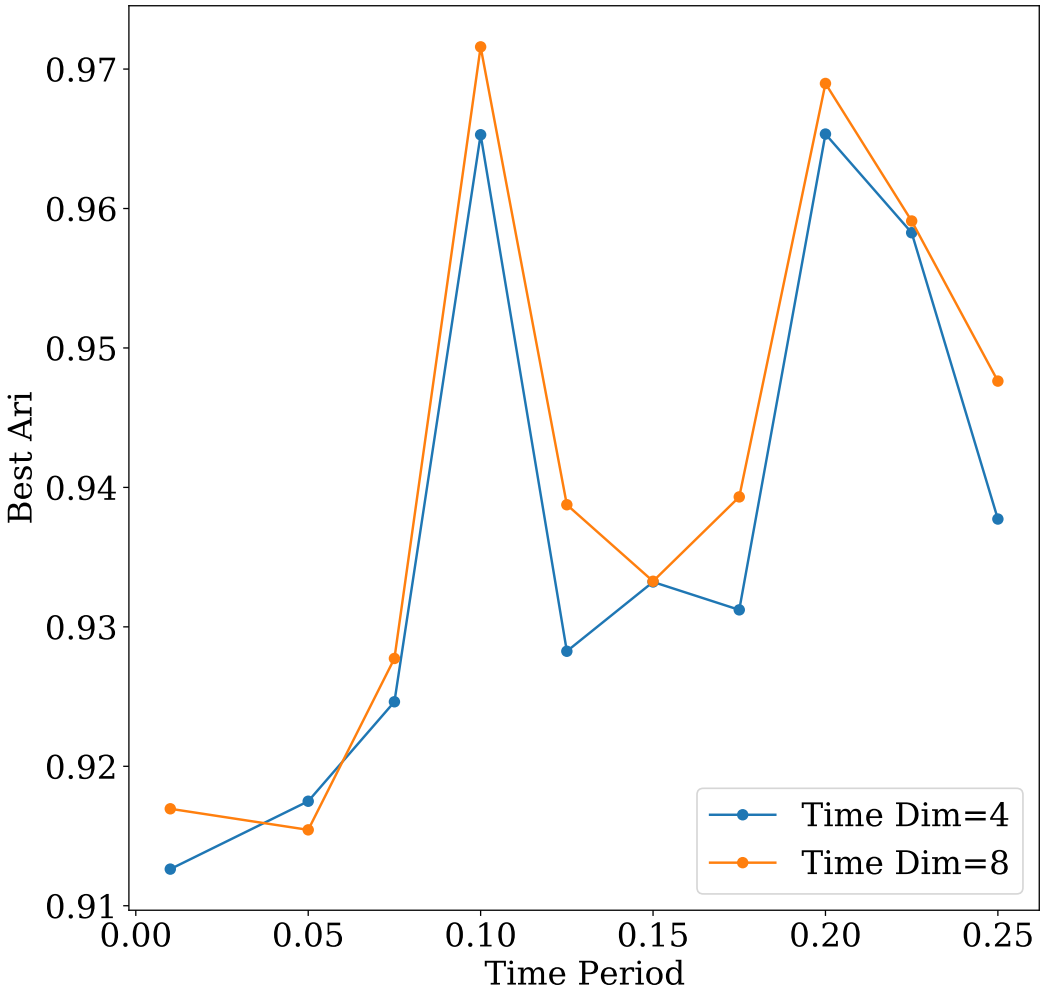


Figure 3: Test ARI for different values of the time dimension K and the characteristic period T.

Experimental Setup We simulate 5 full periods, corresponding to 10 intervals. The model is trained on the first 4 periods (8 intervals) and evaluated on the last (2 intervals). We use an 8-dimensional latent space and sinusoidal time encodings with $K = 16$ features.

Effect of the time dimension and characteristic period We conduct a hyperparameter sweep over the number of time encoding features K and the characteristic period T of the sinusoidal encodings. The results are shown on Figure 3. It can be observed that the best test ARIs are obtained when K is sufficiently large (e.g. $K \geq 8$) and when the characteristic period T is close to the true period of the community structure (i.e. $T = 0.1$).

To assess representation quality, we cluster node embeddings at each queried time step using K-Means and compare the inferred clusters with the ground-truth communities via the Adjusted Rand Index (ARI). The ARI corrects the Rand Index (RI) for chance, where

$$\text{ARI} = \frac{\text{RI} - \text{ExpectedRI}}{\text{MaxRI} - \text{ExpectedRI}},$$

with $\text{MaxRI} = 1$. We report the ARI averaged over time for different parameters in Figure 3, and visualize the learned embeddings via a 1D PCA projection in Figure 1d.

Effect of the time period In this experiment, we report the test ARI for different value for the characteristic period T and the time dimension K. The results are shown on Figure 3. It can be observed that the best test ARIs are observed when the characteristic period is set to 0.1, which is the true period of the community structure.

B.3 Communities Merging over Time

Data Generation We generate a synthetic dataset where two communities merge and split over time by varying the interaction rates between and within communities. The time horizon is $T = 1$ with nodes statically assigned to two communities.

The inter-community and intra-community rates are set as follows:

- On intervals $[0, 0.45]$ and $[0.55, 1]$ (separation periods): intra-community rate $= 4.75$, inter-community rate $= 0.25$
- On interval $[0.45, 0.55]$ (merging period): both rates $= 2.5$

During the merging interval, the community structure effectively disappears as both rates become equal, causing communities to merge. During separation intervals, the large difference in rates maintains distinct community boundaries.

Experimental Setup For each interval and its corresponding rate structure, we sample interactions between node pairs (i, j) from a Poisson distribution with the appropriate rate, then draw interaction times uniformly within the interval. We concatenate interactions across all intervals to obtain the final dataset.

We assess the model's ability to capture the merging and splitting dynamics by visualizing the learned embeddings via 1D PCA projections, which should show nodes separating into clusters during separation periods and merging into a single cluster during the merging interval.

C Declaration of LLM Use

LLMs were used to implement the methods described in this paper. Specifically, ChatGPT and github Copilot were used to help write and debug code for the implementation of the model and experiments. ChatGPT was used to improve the clarity and conciseness of the writing in the paper.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the contributions of the paper, including the introduction of TGAE and its advantages over existing methods.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The paper acknowledges the limitations of TGAE in the conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: The paper describes the model architecture and training procedure, but does not provide details on hyperparameter settings or random seeds used in the experiments. However, the authors commit to releasing the code and data upon acceptance, which will facilitate reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Upon acceptance, the authors will release the code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [No]

   Justification: The paper describes the model architecture and training procedure, but does not provide details on hyperparameter settings or random seeds used in the experiments. However, the authors commit to releasing the code and data upon acceptance, which will facilitate reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not provide error bars or any statistical significance tests for the reported results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

Justification: The paper does not provide detailed information on the compute resources used for the experiments, such as the type of hardware, memory requirements, or time taken for execution.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [NA]

Justification: The research does not involve human subjects or sensitive data, and there are no ethical concerns related to the methods or applications discussed in the paper.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper does not explicitly discuss societal impacts, either positive or negative.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: The paper does not discuss any specific safeguards for the release of its models or data.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the libraries and datasets used in the experiments are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The paper does not provide sufficient documentation for the new assets introduced.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: The paper does not include the full text of instructions given to participants or details about compensation.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The paper does not discuss potential risks to participants or mention any IRB approvals.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs were used to implement the methods described in this paper. Specifically, ChatGPT and github Copilot were used to help write and debug code for the implementation of the model and experiments. ChatGPT was used to improve the clarity and conciseness of the writing in the paper.

    Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.