

# Temporal receptive field in dynamic graph learning: A comprehensive analysis

Yannis Karmim<sup>1</sup>, Leshanshui Yang<sup>3,4</sup>, Raphaël Fournier S'niehotta<sup>1</sup> (✉),  
Nicolas Thome<sup>2</sup>, Clément Chatelain<sup>5</sup>, and Sébastien Adam<sup>3</sup>

<sup>1</sup> Conservatoire National des Arts et Métiers, Paris 75003, France  
`{yannis.karmim,fournier}@cnam.fr`

<sup>2</sup> Sorbonne Université Sciences, Paris 75005, France  
`nicolas.thome@isir.upmc.fr`

<sup>3</sup> Univ Rouen Normandie, LITIS UR 4108, F-76000 Rouen, France  
`clement.chatelain@insa-rouen.fr`

<sup>4</sup> Saagie, 72 Rue de la République, 76140 Le Petit-Quevilly, France  
`leshanshui.yang@saagie.com`

<sup>5</sup> INSA Rouen Normandie, LITIS UR 4108, F-76000 Rouen, France  
`sebastien.adam@univ-rouen.fr`

**Abstract.** Dynamic link prediction is a critical task in the analysis of evolving networks, with applications ranging from recommender systems to economic exchanges. However, the concept of the temporal receptive field, which refers to the temporal context that models use for making predictions, has been largely overlooked and insufficiently analyzed in existing research. In this study, we present a comprehensive analysis of the temporal receptive field in dynamic graph learning. By examining multiple datasets and models, we formalize the role of temporal receptive field and highlight their crucial influence on predictive accuracy. Our results demonstrate that appropriately chosen temporal receptive field can significantly enhance model performance, while for some models, overly large windows may introduce noise and reduce accuracy. We conduct extensive benchmarking to validate our findings, ensuring that all experiments are fully reproducible. Code is available at <https://github.com/ykrmm/BenchmarkTW>.

**Keywords:** Dynamic Link Prediction · GNNs · Evaluation .

## 1 Introduction

Dynamic graphs play an essential role in modeling evolving interactions between entities across various domains, from social networks to computational biology [6,10]. The task of link prediction on such graphs is essential, with numerous applications including forecasting user behavior in recommendation systems, predicting financial transactions, and identifying potential collaborations in academia. Dynamic graphs in these domains are often represented as either a sequence of static graphs captured at regular intervals, referred to as Discrete Time Dynamic

Graphs (DTDG) [26,32], or as Continuous Time Dynamic Graphs (CTDG) which capture interactions as they occur over continuous time [22,24,28,31].

Models that operate on Discrete Time Dynamic Graphs (DTDGs) usually employ a combination of Message-Passing Graph Neural Networks (MP-GNNs) to capture spatial dependencies and Recurrent Neural Networks (RNNs) to capture temporal dependencies [34,19,25,3]. By capturing the evolution of graph structures over time and integrating this dynamic information, these models predict future links with greater accuracy and robustness, addressing the inherent complexities of evolving networks.

However, real-world dynamic graph data come in diverse forms and span different numbers of snapshots. The critical factor is the "length" of dependencies to consider in terms of the number of time steps. This variability poses significant challenges for modeling and prediction, as the number of relevant snapshots can greatly affect the performance of predictive models [15,38,2].

Some models propose to use a temporal window as a hyper-parameter to address this issue [15]. As illustrated in fig. 1, a temporal window defines the span of past data considered for making predictions. Results have shown that in certain cases, a large temporal window can introduce noise and degrade performance. For example, some models like TADDY [15] or Todyformer [2] demonstrate that excessive temporal data can lead to overfitting or irrelevant information overshadowing crucial patterns.

Beyond these marginal hyper-parameter searches in a limited number of models [15], there is a notable lack of a comprehensive analysis on the impact of the temporal window across the wide range of datasets used by the "learning on dynamic graphs" community. This gap calls for a systematic evaluation to understand how different temporal contexts affect dynamic link prediction performance.

In this work, we conduct a comprehensive analysis of the impact of the temporal receptive field on dynamic link prediction performance. We evaluate multiple Discrete Time Dynamic Graph Neural Networks (DTDGNNs) across diverse datasets, examining how different temporal receptive fields influence predictive accuracy. Our contributions are as follows:

- **Temporal receptive field analysis:** We formalize the concept of temporal receptive fields in dynamic graph learning and examine their influence on predictive accuracy and model performance across various scenarios.
- **Model and dataset evaluation::** By comparing the performance of multiple DTDGNNs on a diverse set of datasets, we identify specific contexts in which different temporal window lengths optimize model performance. This analysis provides insights into which datasets benefit most from dynamic information and guides the selection of optimal temporal receptive field.
- **Benchmarking and reproducibility:** We conduct extensive benchmarking by running various DTDG models across multiple datasets. All the experiments in this work are fully reproducible. We provide full access to the code, models, configuration files, and datasets. Our resources ensure that other

researchers can easily integrate new datasets and models, facilitating rapid evaluation of the need for temporal context in dynamic link prediction.

## 2 Context and related work

This section reviews a few key concepts relevant to the dynamic link prediction task with discrete time dynamic graph neural networks.

### 2.1 Discrete Time Dynamic Graph Link Prediction

A Discrete Time Dynamic Graph (DTDG) can be viewed as a sequence of static graphs (called snapshots) as shown in Equation 1 and Figure 1 (a). Each snapshot  $G^t$  is represented by a set of nodes  $V^t$  and a set of edges  $E^t$ , with optional  $\mathbf{X}_v^t$  or  $\mathbf{X}_e^t$  representing node or edge attributes [26,36].  $N^t = |V^t|$  and  $M^t = |E^t|$  indicate respectively the numbers of nodes and edges in snapshot  $G^t$ .

$$DTDG = (G^1, G^2, \dots, G^T) \quad (1)$$

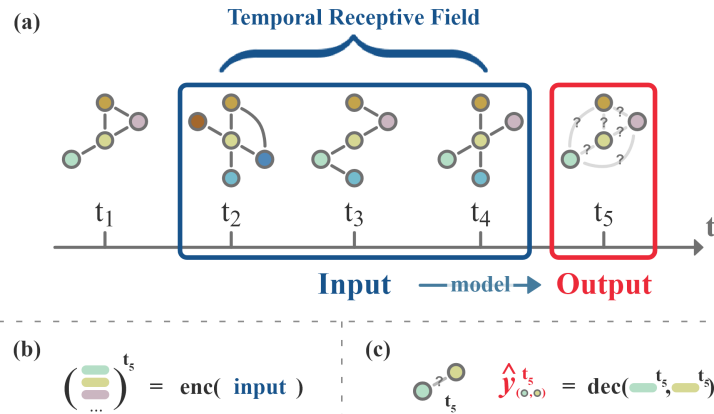


Fig. 1: (a) Dynamic Link Prediction task on a Discrete Time Dynamic Graph: In this figure, the model takes snapshots from  $t_2$  to  $t_4$  as input and predicts the existence of edges at time  $t_5$ . The number of time steps in which the model can capture information is called the size of the receptive field, denoted as  $\tau$ .

(b) When predicting, for example at  $t_5$ , the encoder of the model computes node representation vectors based on the input.

(c) For each edge to be predicted, the decoder of the model computes the relevant node representations to obtain the probability of the existence of the edge. The figure shows the edge between the green and yellow nodes at time  $t_5$  as an example.

The dynamic link prediction task aims to predict the existence of an edge  $e_{u,v}^{t+1}$  between two nodes  $u$  and  $v$  at time  $t+1$  using past information from a time-window of size  $\tau$   $\{G^{t-\tau+1}, \dots, G^t\}$ . A model learns to represent nodes and/or edges with the first  $T_{train}$  snapshots of the DTDG. Then, for each snapshot  $G^t$  in the subsequent  $T_{test}$  time steps, the model predicts the probability of the existence of the edges in the future time step  $t+1$ .

## 2.2 Discrete Time Dynamic Graph Neural Networks

To address DTDG tasks such as Dynamic Link Prediction, encoder-decoder structures are widely used in Discrete Time Dynamic Graph Neural Networks (DTDGNNs) [14,3,11]. The encoder transforms input signals such as attributes and structure of the graph into latent representations. This process is also known as representation learning. The node representations are denoted as  $\mathbf{Z}_v^t \in \mathbb{R}^{N^t \times d}$ , where  $d$  represents the size of the vector, as shown in figure 1 (b). Edges can be encoded in the same way [23,33]. The decoder, coming after the encoder in a neural network, computes latent representations to obtain predictions for downstream tasks [8], as shown in figure 1 (c).

To handle both temporal and structural information, the DTDG encoder typically combines static graph encoders  $f_G(\cdot)$  with temporal encoders  $f_T(\cdot)$  [32]. Graph encoders commonly rely on Graph Neural Networks (GNN) [30], Random Walk-based methods [7], or Graph Transformers [17] for propagating node representations and capturing structural information within a snapshot. Temporal encoders typically leverage Recurrent Neural Networks (RNN) [9], Temporal Convolutional Networks (TCN) [18], or Transformers [27] for capturing temporal patterns among multiple time steps.

From the perspective of how to encode information across multiple snapshots, DTDGNNs fall into two main paradigms: Sequentially Encoding Hidden states and Sequentially Encoding Model Parameters, denoted as **Enc(H)** and **Enc(Θ)** respectively, as shown in figure 2.

The **Enc(H)** paradigm uses the graph encoder  $f_G(\cdot)$  to encode each snapshot as a static graph, and then uses the temporal encoder  $f_T(\cdot)$  to encode temporal information across snapshots, as shown in equation 2. The two modules can be combined in either a stacked or integrated structure. In stacked structures,  $f_G$  and  $f_T$  layers are stacked one or several times. For example, **DySAT** [25] successively uses structural self-attention [27] and temporal self-attention to encode graph and temporal information; **STGCN** [35] uses stacked (TCN, GNN, TCN) layers as the basic module and repeats it several times.

**Enc(H)** with integrated structure takes another approach by incorporating  $f_G(\cdot)$  into  $f_T(\cdot)$ . Given that  $f_T(\cdot)$  has linear projection [9] or convolution modules [18] to handle the node features, this approach aggregates neighboring features by replacing these modules with  $f_G(\cdot)$ . Typical models include **GC-LSTM** [3] replacing the linear layer in LSTM [9] with GCN [12].

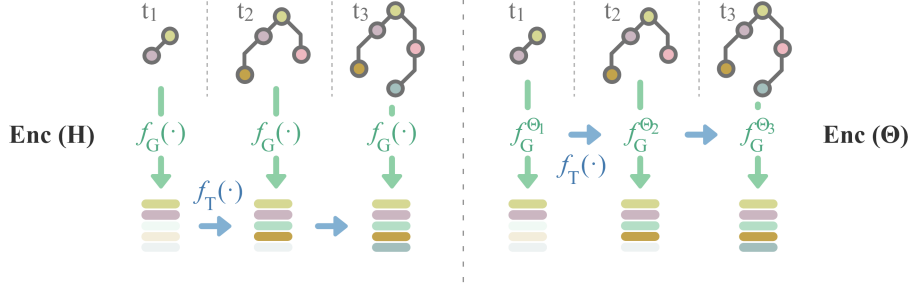


Fig. 2: Two categories of Discrete Time Dynamic Graph Neural Networks (DT-DGNNs). Left: Sequentially encoding the hidden states  $\mathbf{H}$  of each snapshot across time with a temporal encoder  $f_T(\cdot)$ . Right: Sequentially encoding the parameters  $\Theta$  of the graph encoder  $f_G(\cdot)$  across time with a temporal encoder  $f_T(\cdot)$ .

$$\begin{aligned} \mathbf{H}^k &= f_G(\mathbf{A}^k, \mathbf{H}^k), \quad k \in [t - \tau + 1, t] \\ \mathbf{H}^{t+1} &= f_T(\mathbf{H}^{t-\tau+1:t}) \end{aligned} \quad (2)$$

Another paradigm  $\mathbf{Enc}(\Theta)$  lets the parameters of  $f_G(\cdot)$  keep evolving. This paradigm uses  $f_T(\cdot)$  to encode the parameters of  $f_G(\cdot)$  across snapshots, allowing each snapshot to be encoded in a more adaptive way. **EvolveGCN** (EGCN) [19] suggests dynamically updating the GCN [12] weight parameters. As shown in equation 3, the graph encoder parameters  $\Theta_{f_G}^{t+1}$  in each snapshot are predicted by a temporal encoder.

$$\Theta_{f_G}^{t+1} = f_T(\Theta_{f_G}^t, \mathbf{H}^t) \quad \mathbf{H}^{t+1} = f_G(\mathbf{A}^t, \mathbf{H}^t; \Theta_{f_G}^{t+1}) \quad (3)$$

The diverse model architectures demonstrate how existing research incorporates temporal contexts when extending static graph encoders to dynamic graphs. This opens up possibilities for various tasks and applications of DTDG.

### 3 Temporal Receptive Field in Dynamic Graph Representation Learning

Capturing temporal patterns plays an important role in Dynamic Graph Representation Learning. Toward better temporal encoding, existing research has primarily focused on structural optimizations of  $f_T(\cdot)$ , such as incorporating attention mechanisms [25] or gating mechanisms [14,35]. However, there has been limited discussion of the temporal receptive field of DTDGs.

We introduce the concept of the **temporal receptive field**. The temporal receptive field refers to the temporal information or contexts in the data that

a model can capture and use when processing sequential data [29,21]. Different neural network architectures handle temporal contexts in various ways.

Recurrent Neural Networks (RNN) [9,4] process sequential data through a recursive structure, where the output of each time step depends not only on the current input but also on the hidden state from the previous time step. Due to this recursive nature, each step connects to the entire time steps in the given input, theoretically making its temporal receptive field encompass all input time steps. Practically, vanilla RNNs are trained using the Backpropagation Through Time algorithm that unfolds the network, making it deeper and subject to the vanishing gradient issue [20]. Therefore, the unfolded network is generally limited to a reduced number of time steps, thus limiting their temporal receptive field. LSTM and GRUs partially compensate for this limitation by adding an explicit memory cell in their units, but can still suffer from the vanishing gradient.

Among non-RNN structures, common temporal encoders include Temporal Convolutional Networks (TCNs) and Transformers. TCNs apply causal convolution [18] along time steps to process sequential data. Transformers utilize self-attention mechanisms [27] to process sequential data, allowing each input position to interact with all other positions in the sequence. The size of their receptive field  $\tau$  is explicitly determined by model hyper-parameters.

Intuitively, a larger temporal receptive field contains richer past information for the model to capture. However, in practice, RNNs suffer from the problem of vanishing or exploding gradients [20]. More complex RNN models such as LSTM and GRU also have problems with the effectiveness of memory over long sequences [1,37]. Empirically, some models indicate that incorporating more time steps in dynamic graphs does not necessarily yield better performance [15,2].

To our knowledge, the influence of the temporal receptive field size on model performance on DTDGs remains an unexplored area. Many analyses of the temporal receptive field of the inputs/outputs in DTDGs focus on hyper-parameter search [15,11,5,38], highlighting the necessity of this research. This emphasizes the importance of a benchmark in this field to investigate how much temporal context is necessary to perform effectively on dynamic link prediction tasks, and extend to generalized DTDG representation learning.

## 4 Experiments

The dynamic link prediction task is often modeled as a binary classification problem, where truly existing edges labeled as 1 and non-existent edges labeled as 0 [19,3,11,25]. As a standard practice, existing studies use the "negative sampling" method when evaluating model performance [19]. This method samples  $M^{t+1}$  non-existent edges and mixes in  $M^{t+1}$  truly existing edges at  $t+1$  to formulate a balanced binary classification task. The average precision is therefore widely used to evaluate the performance of the model. This is also the case in our experiments.

The goal of these experiments is to address the following questions: (1) How does the temporal receptive field influence results in dynamic link prediction (section 4.2) ? (2) Which datasets require the broadest temporal receptive field

(section 4.3) ? (3) What is each model’s capacity to capture dynamic information (section 4.4) ? Additionally, these experiments provide a rigorous benchmarking of several models across a wide range of datasets used by the dynamic graph learning community. At the end of this experimental section, we also discuss the implications of the results for DTDGNN models and suggest future research directions (section 4.5).

#### 4.1 Baselines and Datasets

Table 1: Statistics of datasets used in our experiments.

Datasets	Domains	Nodes	Links	Snapshots	Duration
CanParl	Politics	734	74,478	14	14 years
USLegis	Politics	225	60,396	12	12 congresses
Trade	Economics	255	507,497	32	32 years
UNVote	Politics	201	1,035,742	72	72 years
UCI-Message	Social	1,899	59,835	88	196 days
AS733	Router	6,628	13,512	30	86 days
Enron	Mail	184	790	11	3 years
Colab	Citations	315	943	10	9 years
Bitcoin-OTC	Trust Networks	5,881	35,592	136	5 years
Bitcoin-Alpha	Trust Networks	3,783	24,186	136	5 years

To answer these questions, we selected ten discrete dynamic graph datasets (table 1) commonly used by the dynamic graph learning community [19,25,22]. These graphs are diverse, ranging from social networks to economic and political networks. The datasets CanParl, USLegis, Trade, and UNVote are sourced from the dynamic link prediction evaluation paper [22]. The UCI-Message, AS733, Bitcoin OTC, and Bitcoin Alpha datasets were used by the EGCN model [19] and are also part of the Stanford database [13]. We also included the Enron email dataset and the scientific collaboration dataset Colab from [13]. An initial observation is the disparity in the lifespan of these DTDGs, ranging from a few days for the AS733 router network to almost a century for the UNVote dataset. We provide more details of each dataset in section B of the appendix.

#### 4.2 Enhancing dynamic link prediction with an optimal temporal receptive field

This subsection and results address the question: *(1)How does the temporal receptive field influence results in dynamic link prediction?* Our first results in fig. 3 demonstrate that adapting the temporal receptive field  $\tau$  can significantly enhance model performance. The first part of the table is the score of models using all the temporal information, with a temporal receptive field of  $\tau_\infty$ . This is

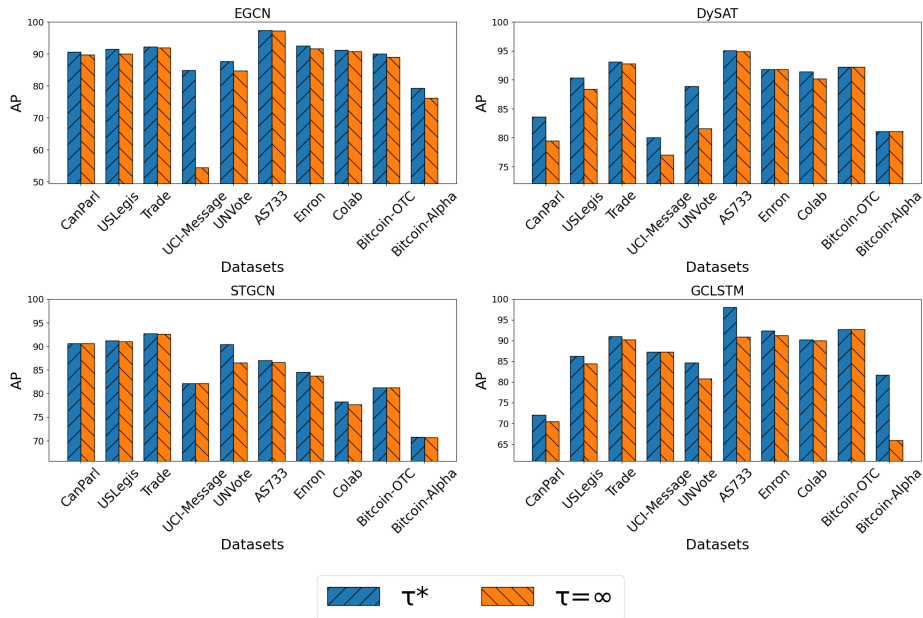


Fig. 3: Model performance in average precision, with an optimal temporal receptive field  $\tau^*$  vs. all temporal information  $\tau_\infty$ .

the standard implementation of existing DTDGNNs models [25,35,19,3]. To find the optimal value of  $\tau$ , we conducted a parameter search within a range from 1 to the number of snapshots. The results for the optimal  $\tau$  are shown in the second part of the table  $\tau^*$ . The optimal  $\tau$  value is indicated in parentheses following the AP score. Using the optimal temporal receptive field  $\tau^*$  generally improves performance compared to using all temporal information  $\tau_\infty$ . For instance, **EGCN** shows an average gain of +3.40 points in AP, while **GCLSTM** improves by +3.24 points, **DySat** +1.80, **Edgebank** +1.92 and 0.58 for **STGCN**. These findings highlight the importance of selecting an appropriate  $\tau$  tailored to the models and the dataset’s characteristics.

For most datasets, using the optimal temporal receptive field  $\tau^*$  leads to marked improvements in performance compared to using all temporal information  $\tau_\infty$ . This indicates that carefully selecting a temporal window size can enhance predictive accuracy significantly. We present more detailed results in table 4 of section A of the appendix.

### 4.3 Analysis of temporal receptive field size and dataset characteristics

This subsection addresses the question (2): *Which datasets require the broadest temporal receptive field?* In the table 2, we show that for some datasets, using the



$\tau$	Datasets	EGCN [19]	DySat [25]	GCLSTM [3]	STGCN [35]	EdgeBank [22]
$\tau_\infty$	CanParl	89.70	<b>79.44</b>	<b>70.47</b>	90.64	<b>54.48</b>
	USLegis	90.02	<b>88.36</b>	<b>84.42</b>	<b>90.95</b>	57.75
	Trade	91.95	92.78	90.15	92.62	65.94
	UCI-Message	54.33	<b>77.04</b>	<b>87.22</b>	<b>82.14</b>	<b>76.28</b>
	UNVote	84.76	81.60	<b>80.75</b>	86.49	61.17
	AS733	97.17	<b>94.88</b>	90.86	86.66	<b>97.21</b>
	Enron	91.58	<b>91.84</b>	<b>91.26</b>	83.73	<b>79.35</b>
	Colab	90.69	90.20	<b>89.99</b>	<b>77.68</b>	<b>77.02</b>
	Bitcoin-OTC	89.04	<b>92.23</b>	<b>92.65</b>	<b>81.29</b>	<b>52.18</b>
	Bitcoin-Alpha	<b>76.19</b>	<b>81.10</b>	65.91	<b>70.72</b>	<b>57.43</b>
$\tau_1$	CanParl	<b>90.56</b>	73.02	53.03	90.65	51.37
	USLegis	<b>90.74</b>	83.94	73.33	90.25	73.46
	Trade	<b>92.12</b>	<b>93.07</b>	<b>90.76</b>	<b>92.70</b>	<b>82.42</b>
	UCI-Message	<b>84.85</b>	66.91	85.22	81.04	55.06
	UNVote	<b>86.96</b>	<b>83.93</b>	74.43	<b>87.89</b>	<b>97.30</b>
	AS733	<b>97.37</b>	94.57	<b>98.03</b>	<b>86.94</b>	82.15
	Enron	<b>92.45</b>	91.28	89.29	<b>84.49</b>	71.53
	Colab	<b>90.86</b>	<b>91.39</b>	89.44	76.89	70.34
	Bitcoin-OTC	<b>89.99</b>	60.32	87.55	79.99	50.08
	Bitcoin-Alpha	73.51	54.78	<b>81.70</b>	67.00	51.33

Table 2: Average precision of DTDG models on multiple datasets using all temporal information  $\tau_\infty$  vs using no dynamic information  $\tau_1$ .

last snapshot alone (i.e  $\tau = 1$ ) yields high predictive performance across all models. For the **UNVote** dataset, most models also achieve better predictions without using explicitly the dynamic evolution in the input structure. This is particularly notable with the non-parametric **EdgeBank** baseline, where predicting the same links as in the previous time step almost reaches perfect results. This can be explained by the nature of the dataset. The **UNVote** dataset documents roll-call votes conducted in the United Nations General Assembly. Whenever two nations cast a "yes" vote for an item, a link is established between them. Often, the policies of two countries are aligned within a short time frame, so looking at the relationships from 70 years ago is not particularly helpful. For the **Trade** dataset, we observe a similar phenomenon for all models. The **Trade** dataset covers trade in food and agriculture products between 181 nations over more than 30 years, with links reflecting the cumulative sum of normalized import or export values. Short-term trading patterns are more relevant, making older data less useful for predictions.

In contrast, for the **Bitcoin-OTC** and **Bitcoin-Alpha** datasets, users' reputations must be tracked over time to avoid fraudulent transactions, making long-term interactions crucial. In the **UCI-Message** dataset, although the duration is only 196 days, social interactions and message exchanges evolve rapidly, necessitating a broader temporal context to capture meaningful patterns and

trends.

However, the table 2 also shows that performance does not solely depend on the datasets but also on the model’s ability to capture long-term dynamic information. For example, with EGCN, we observe that performance with a receptive field of  $\tau_1$  is almost always better. This leads us to the next subsection and the need to analyze the different models in more depth.

#### 4.4 Analysis of the models’ ability to capture dynamic information.

This subsection addresses the question 3: *What is each model’s capacity to capture dynamic information?* To do so, we propose two analyses.

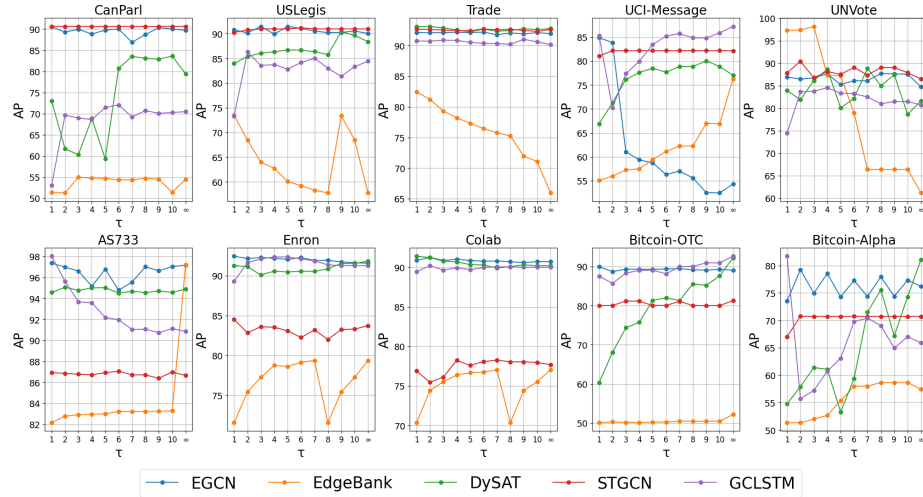


Fig. 4: Average precision (AP) scores of various DTDG models across multiple datasets, shown as a function of the temporal receptive field  $\tau$ . A value of  $\tau_\infty$  represents the use of all temporal information.

**a.) Performance of models in link prediction with varying temporal receptive fields  $\tau$  (fig. 4)** The figure fig. 4 presents the average precision scores of various DTDG models across multiple datasets as a function of the temporal receptive field  $\tau$ . For most datasets, the performance of the **EGCN** model remains relatively better with smaller value of  $\tau$ , with a noticeable drop for larger temporal receptive field in some cases, such as Trade and UNVote. This suggests that **EGCN** is not able to capture long-term dynamic information. The **DySAT** model shows more variability across datasets, but performs better with larger temporal receptive field. For USLegis and UNVote, performance improves

a lot with larger temporal windows, indicating the ability of capturing dynamic information.

**GCLSTM** generally benefits from larger temporal receptive field, especially in the UCI-Message dataset, in section 4.3 we underline the importance of long temporal context in this dataset. **STGCN** is relatively stable as the temporal receptive field increases. Except UNVote and Trade for the reason that we explain in section 4.3, **EdgeBank** is better when it consider all temporal information.

**b.) Correlation between the number of snapshots and the AP score in dynamic link prediction (table 3)** To evaluate the models’ ability to capture long-term information, we calculated the correlation between the difference in performance at  $\tau_\infty$  and  $\tau_1$  and the number of snapshots. This correlation analysis in table 3 reveals that models such as **DySAT** and **STGCN** exhibit a positive correlation with the total number of snapshots, demonstrating their ability to capture long-term temporal information. Conversely, **EGCN** and **GCLSTM** show negative correlations, indicating a reduced capacity to leverage long-term dynamic information effectively.

Model	Correlation with total number of snapshots
DySAT	0.85
EGCN	-0.19
GCLSTM	-0.43
STGCN	0.62
EdgeBank	0.09

Table 3: Correlation between the number of snapshot and the difference in model performances ( $\tau_\infty - \tau_1$ ).

#### 4.5 Discussion and future research direction

These results highlight the necessity of analyzing the real capacity of dynamic graph learning models to capture temporal information. It is also essential to carefully select datasets to determine if a dynamic model is truly needed for future predictions. We demonstrated that several standard DTDGNNs models in the dynamic community are not consistently better at capturing temporal information compared to predicting solely from the previous snapshot (table 2). We also observed a lack of robustness concerning the size of the temporal receptive field (fig. 4) and the inability of some models to handle dynamic information over long data sequences (table 3). Additionally, we found that a judicious choice of an optimal temporal context can significantly improve the performance of certain models (fig. 3). Future research can further explore the effects of varying the temporal receptive field size across different categories of models [32,16], such

as parameter-encoding models like EvolveGCN [19], those that focus only on hidden state encoding [35,3], as well as extending this analysis to more recent architectures such as Dynamic Graph Transformers. These findings raise an open question: Is the task itself not requiring temporal context, or are the models struggling to capture this information effectively? This is a crucial question that future research must address. The varied behavior of the tested models suggests that this issue needs to be examined across a broader range of datasets and models. More comprehensive evaluations, such as those focusing on new links and historical links, as suggested in recent works [22] will be essential to gain deeper insights. We encourage the dynamic graph learning community to assess the true dynamic capability of models by integrating the proposed evaluations.

## 5 Conclusion

In this work, we formalize the concept of the temporal receptive field and conduct an in-depth evaluation of it in the task of link prediction on DTDGs. We revealed the capacity of each model to capture dynamic information and pinpointed the datasets that require a long temporal context. Our analyses showed that having a very large temporal receptive field is surprisingly not synonymous with better predictions, and sometimes, not considering past information leads to better results.

**Disclosure of Interests.** All authors disclosed no relevant relationships.

## References

1. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
2. Biparva, M., Karimi, R., Faez, F., Zhang, Y.: Todyformer: Towards holistic dynamic graph transformers with structure-aware tokenization (2024)
3. Chen, J., Wang, X., Xu, X.: Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction. *Applied Intelligence* pp. 1–16 (2022)
4. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
5. Fan, J., Bai, J., Li, Z., Ortiz-Bobea, A., Gomes, C.P.: A gnn-rnn approach for harnessing geospatial and temporal information: application to crop yield prediction. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 36, pp. 11873–11881 (2022)
6. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation (2 2019), <http://arxiv.org/abs/1902.07243>, recommendation
7. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 855–864 (2016)
8. Hamilton, W.L.: *Graph representation learning*. Morgan & Claypool Publishers (2020)
9. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. *Neural Computation* **9**(8), 1735–1780 (11 1997). <https://doi.org/10.1162/NECO.1997.9.8.1735>
10. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstern, S., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P., Hassabis, D.: Highly accurate protein structure prediction with AlphaFold. *Nature* 2021 596:7873 **596**(7873), 583–589 (7 2021). <https://doi.org/10.1038/s41586-021-03819-2>, <https://www.nature.com/articles/s41586-021-03819-2>
11. King, I.J., Huang, H.H.: Euler: Detecting network lateral movement via scalable temporal link prediction. *ACM Transactions on Privacy and Security* **26**(3), 1–36 (2023)
12. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. arXiv preprint arXiv:1609.02907 (9 2016), <http://arxiv.org/abs/1609.02907>
13. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data> (Jun 2014)
14. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
15. Liu, Y., Pan, S., Wang, Y.G., Xiong, F., Wang, L., Chen, Q., Lee, V.C.: Anomaly Detection in Dynamic Graphs via Transformer. *IEEE Transactions on Knowledge and Data Engineering* **35**(12), 12081–12094 (2021). <https://doi.org/10.1109/TKDE.2021.3124061>
16. Longa, A., Lachi, V., Santin, G., Bianchini, M., Lepri, B., Lio, P., Scarselli, F., Passerini, A.: Graph neural networks for temporal graphs: State of the art, open challenges, and opportunities. arXiv preprint arXiv:2302.01018 (2023)

17. Min, E., Chen, R., Bian, Y., Xu, T., Zhao, K., Huang, W., Zhao, P., Huang, J., Ananiadou, S., Rong, Y.: Transformer for graphs: An overview from architecture perspective. arXiv preprint arXiv:2202.08455 (2022)
18. Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)
19. Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C.: Evolvegcnn: Evolving graph convolutional networks for dynamic graphs. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 5363–5370 (2020)
20. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318. Pmlr (2013)
21. Peddinti, V., Povey, D., Khudanpur, S.: A time delay neural network architecture for efficient modeling of long temporal contexts. In: Interspeech. pp. 3214–3218 (2015)
22. Poursafaei, F., Huang, S., Pelrine, K., Rabbany, R.: Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems* **35**, 32928–32941 (2022)
23. Renton, G.: Analyzing and improving graph neural networks. Theses, Normandie Université (Jul 2021), <https://theses.hal.science/tel-03346018>
24. Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., Bronstein, M.: TGN: Temporal Graph Networks for Deep Learning on Dynamic Graphs. arXiv preprint arXiv:2006.10637 (6 2020), <https://arxiv.org/abs/2006.10637>
25. Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H.: Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In: Caverlee, J., Hu, X.B., Lalmas, M., Wang, W. (eds.) *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining*, Houston, TX, USA, February 3–7, 2020. pp. 519–527. ACM (2020). <https://doi.org/10.1145/3336191.3371845>, <https://doi.org/10.1145/3336191.3371845>
26. Skarding, J., Gabrys, B., Musial, K.: Foundations and Modeling of Dynamic Networks Using Dynamic Graph Neural Networks: A Survey. *IEEE Access* **9**, 79143–79168 (2021). <https://doi.org/10.1109/ACCESS.2021.3082932>
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. arXiv preprint arXiv:1706.03762 (6 2017), <https://arxiv.org/abs/1706.03762>
28. Wang, Y., Chang, Y.Y., Liu, Y., Leskovec, J., Li, P.: Inductive representation learning in temporal networks via causal anonymous walks. In: *ICLR 2021 - 9th International Conference on Learning Representations* (1 2021). <https://doi.org/10.48550/arxiv.2101.05974>, <http://arxiv.org/abs/2101.05974>
29. Wang, Y., Li, H.: Industrial process time-series modeling based on adapted receptive field temporal convolution networks concerning multi-region operations. *Computers & Chemical Engineering* **139**, 106877 (2020). <https://doi.org/https://doi.org/10.1016/j.compchemeng.2020.106877>, <https://www.sciencedirect.com/science/article/pii/S0098135419313109>
30. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* **32**(1), 4–24 (2020)
31. Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K.: TGAT: Inductive Representation Learning on Temporal Graphs. *ICLR 20* (2 2020). <https://doi.org/10.48550/arxiv.2002.07962>, <http://arxiv.org/abs/2002.07962>

32. Yang, L., Chatelain, C., Adam, S.: Dynamic Graph Representation Learning With Neural Networks: A Survey. *IEEE Access* **12**, 43460–43484 (2024). <https://doi.org/10.1109/ACCESS.2024.3378111>
33. Yang, Y., Li, D.: Nenn: Incorporate node and edge features in graph neural networks. In: Pan, S.J., Sugiyama, M. (eds.) *Proceedings of The 12th Asian Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 129, pp. 593–608. PMLR (18–20 Nov 2020), <https://proceedings.mlr.press/v129/yang20a.html>
34. You, J., Du, T., Leskovec, J.: Roland: Graph learning framework for dynamic graphs. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), <https://api.semanticscholar.org/CorpusID:251518327>
35. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017)
36. Zaki, A., Attia, M., Hegazy, D., Amin, S.: Comprehensive survey on dynamic graph models. *International Journal of Advanced Computer Science and Applications* **7**(2) (2016)
37. Zhao, J., Huang, F., Lv, J., Duan, Y., Qin, Z., Li, G., Tian, G.: Do RNN and LSTM have long memory? In: III, H.D., Singh, A. (eds.) *Proceedings of the 37th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 119, pp. 11365–11375. PMLR (13–18 Jul 2020), <https://proceedings.mlr.press/v119/zhao20c.html>
38. Zheng, G., Chai, W.K., Zhang, J., Katos, V.: Vdgcnet: A novel network-wide virtual dynamic graph convolution neural network and transformer-based traffic prediction model. *Knowledge-Based Systems* **275**, 110676 (2023). <https://doi.org/https://doi.org/10.1016/j.knosys.2023.110676>, <https://www.sciencedirect.com/science/article/pii/S0950705123004264>

## A Additional results

This table 4 presents additional results, showing the comparison of Average Precision (AP) scores for various Dynamic Temporal Graph Neural Networks (DTGNNs) across multiple datasets. The results compare using all available temporal information  $\tau_\infty$  versus an optimal temporal receptive field  $\tau^*$ . The average gains in AP scores for each model are also indicated.

$\tau$	Datasets	EGCN [19]	DySat [25]	GCLSTM [3]	STGCN [35]	EdgeBank [22]
$\tau_\infty$	CanParl	89.70	79.44	70.47	90.64	54.48
	USLegis	90.02	88.36	84.42	90.95	57.75
	Trade	91.95	92.78	90.15	92.62	65.94
	UCI-Message	54.33	77.04	87.22	82.14	76.28
	UNVote	84.76	81.60	80.75	86.49	61.17
	AS733	97.17	94.88	90.86	86.66	97.21
	Enron	91.58	91.84	91.26	83.73	79.35
	Colab	90.69	90.20	89.99	77.68	77.02
	Bitcoin-OTC	89.04	92.23	92.65	81.29	52.18
	Bitcoin-Alpha	76.19	81.10	65.91	70.72	57.43
$\tau^*$	CanParl	90.56 (1)	83.63 (10)	72.06 (6)	90.65 (1)	54.93 (3)
	USLegis	90.74 (5)	90.38 (9)	86.26 (2)	90.95 (3)	73.46 (1)
	Trade	92.12 (6)	93.13 (2)	91.02 (9)	92.72 (6)	82.42 (1)
	UCI-Message	84.85 (1)	80.03 (9)	87.22 ( $\infty$ )	82.15 (2)	76.28 ( $\infty$ )
	UNVote	86.96 (4)	88.71 (4)	84.57 (4)	90.42 (2)	97.30 (1)
	AS733	97.37 (1)	95.07 (2)	98.03 (1)	87.06 (6)	97.21 ( $\infty$ )
	Enron	92.45 (1)	91.84 ( $\infty$ )	92.35 (5)	84.49 (1)	79.35 (7)
	Colab	91.01 (4)	91.39 (1)	90.19 (2)	78.25 (7)	77.02 (7)
	Bitcoin-OTC	89.99 (1)	92.23 ( $\infty$ )	92.65 ( $\infty$ )	81.29 ( $\infty$ )	52.18 ( $\infty$ )
	Bitcoin-Alpha	79.25 (2)	81.10 ( $\infty$ )	81.70 (1)	70.75 (2)	58.67 (8)
<b>Avg gain</b>		+3.40	+1.80	+3.24	+0.58	+1.92

Table 4: Comparison of Average precision (AP) scores for various DTGNNs across multiple datasets, using all available temporal information  $\tau_\infty$  versus an optimal temporal receptive field  $\tau^*$ . For  $\tau^*$ , we indicate in parentheses, the optimal value.

## B Datasets

Here, we describe the datasets used in our experiments. These descriptions are sourced from the Stanford SNAP database [13] as well as from the dynamic link prediction evaluation paper [22].

- **CanParl**: Can. Parl. is a network that tracks how Canadian Members of Parliament (MPs) interacted between 2006 and 2019. Each dot represents



an MP, and a line connects them if they both said "yes" to a bill. The line's thickness shows how often one MP supported another with "yes" votes in a year.

- **USLegis**: USLegis is a Senate co-sponsorship network that records how lawmakers in the US Senate interact socially. The strength of each connection indicates how many times two senators have jointly supported a bill during a specific congressional session
- **Bitcoin-OTC**: This is who-trusts-whom network of people who trade using Bitcoin on a platform called Bitcoin OTC. Since Bitcoin users are anonymous, there is a need to maintain a record of users' reputation to prevent transactions with fraudulent and risky users. Members of Bitcoin OTC rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1. This is the first explicit weighted signed directed network available for research [13].
- **Bitcoin-Alpha**: This is who-trusts-whom network of people who trade using Bitcoin on a platform called Bitcoin Alpha. Since Bitcoin users are anonymous, there is a need to maintain a record of users' reputation to prevent transactions with fraudulent and risky users. Members of Bitcoin Alpha rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1. This is the first explicit weighted signed directed network available for research [13].
- **Trade**: UNTrade covers the trade in food and agriculture products between 181 nations over a span of more than 30 years. The weight assigned to each link within this dataset reflects the cumulative sum of normalized import or export values for agricultural goods exchanged between two specific countries.
- **UNVote**: UNVote documents roll-call votes conducted in the United Nations General Assembly. Whenever two nations cast a "yes" vote for an item, the link's weight connecting them is incremented by one.
- **Contact**: Contact dataset provides insights into the evolving physical proximity among approximately 700 university students over the course of a month. Each student is uniquely identified, and links between them indicate their close proximity. The weight assigned to each link reveals the degree of physical proximity between the students
- **Enron**: Enron consists of emails exchanged among 184 Enron employees. Nodes represent employees, and edges indicate email interactions between them. The dataset includes 10 snapshots and does not provide node or edge-specific information
- **Colab**: Colab represents an academic cooperation network, capturing the collaborative efforts of 315 researchers from 2000 to 2009. In this network, each node corresponds to an author, and an edge signifies a co-authorship relationship.
- **UCI-Message**: This dataset is comprised of private messages sent on an online social network at the University of California, Irvine. Users could search the network for others and then initiate conversation based on profile information. An edge  $(u, v, t)$  means that user  $u$  sent a private message to user  $v$  at time  $t$ . The dataset here is derived from the one hosted by Tore

Opsahl, but we have parsed it so that it can be loaded directly into SNAP as a temporal network [13].

### C Comparative Architecture Overview of Dynamic Graph Models

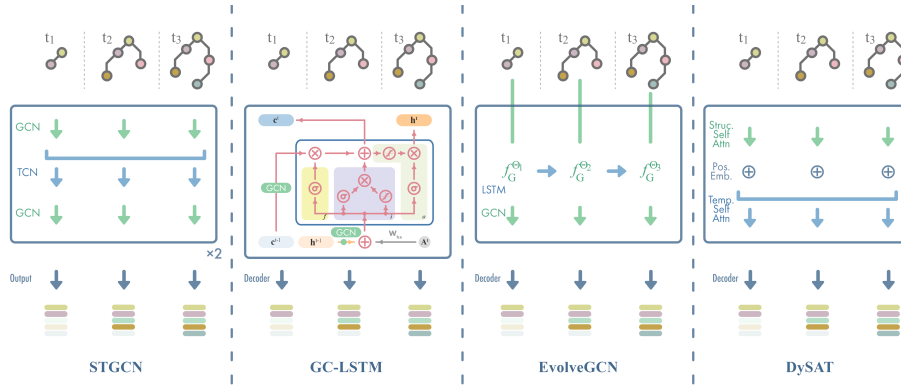


Fig. 5: Schematic representation of the architectural differences among EGCN, Dysat, STGCN, and GCLSTM models. This comparison highlights the varying approaches these models take in encoding temporal and structural information within dynamic graphs.