Temporal Graph Generative Models: An empirical study

Houssem Eddine Souid*
houssem.souid@euranova.eu
Euranova
Tunis, Tunisia

Lucas Ody*
lucas.ody@euranova.eu
Euranova
Mont-Saint-Guibert, Belgium

Youssef Achenchabe*
youssef.achenchabe@euranova.eu
Euranova
Marseille, France

Valentin Lemaire*
valentin.lemaire@euranova.eu
Euranova
Mont-Saint-Guibert, Belgium

Gianmarco Aversano
gianmarco.aversano@euranova.eu
Euranova
Mont-Saint-Guibert, Belgium

Sabri Skhiri
sabri.skhiri@euranova.eu
Euranova
Mont-Saint-Guibert, Belgium

Abstract

Graph Neural Networks (GNNs) have recently emerged as popular methods for learning representations of non-euclidean data often encountered in diverse areas ranging from chemistry to source code generation. Recently, researchers have focused on learning about temporal graphs, wherein the nodes and edges of a graph and their respective features may change over time. In this paper, we focus on a nascent domain: learning generative models on temporal graphs. We have noticed that papers on this topic so far have lacked a standard evaluation for all existing models on the same benchmark of datasets and a solid evaluation protocol. We present extensive comparative experiments on state-of-the-art models from the literature. Furthermore, we propose a rigorous evaluation protocol to assess temporal generation quality, utility, and privacy.

CCS Concepts: • **Systems for machine learning/machine learning for systems**;

Keywords: Dynamic graphs, Synthetic data generation, Evaluation framework

ACM Reference Format:
Houssem Eddine Souid, Lucas Ody, Youssef Achenchabe, Valentin Lemaire, Gianmarco Aversano, and Sabri Skhiri. 2024. Temporal Graph Generative Models: An empirical study. In *4th Workshop on Machine Learning and Systems (EuroMLSys '24), April 22, 2024, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3642970.3655829

*Authors contributed equally to this research.

EuroMLSys '24, April 22, 2024, Athens, Greece

ACM ISBN 979-8-4007-0541-0/24/04
https://doi.org/10.1145/3642970.3655829

1 Introduction

Graphs are ubiquitous data structures encountered in miscellaneous areas: chemistry, biology, social and call networks. They model connections (i.e., edges) between individual units (i.e., nodes), such as social networks (networks of friendships) and molecules (networks of atoms). In recent years, deep generative learning has shown spectacular image and text generation results. Researchers extended these models to graph data [1, 3, 5, 10, 12, 13, 20].

The current deep generative models for static graphs have been extended to dynamic graphs due to their essential role in numerous vital domains, such as anomaly detection [6]. Many techniques have been proposed in the literature [2, 4, 7, 14, 15, 21–23]. Some models consider each single time series as one statistical sample and seek to generate the entire time series in a manner similar to this single realisation. Other models aim to learn how to update a graph snapshot. In other words, they learn a probability distribution $p(G_t|G_{t-1})$ such that given an arbitrary graph G_{t-1}, many samples could be drawn for G_t and forecast the future trajectory of the graph.

We observe in the mentioned papers that there is a lack of a standard benchmark to compare the different generative models for dynamic graphs. Furthermore, we noticed that the set of the chosen metrics to evaluate these models differ between papers, and sometimes, metrics assessing the quality of the temporal consistency of the generated graphs are not considered. In this paper, we propose a new evaluation framework for generative models on dynamic graphs with a set of synthetic and real datasets. In addition, metrics assessing the quality of the generation will be considered to compare the performance of state-of-the-art models.

2 Related work

In the extensive landscape of graph generation evaluation metrics, a predominant focus lies in examining topological

statistics (e.g. node degree distribution, clustering coefficient distribution) across the original and the generated sets. However, this often results in node-wise values. To obtain a single, scalar value, this array would need to be aggregated in both the time and the node dimensions. This review categorises existing techniques based on their underlying assumptions and computation methodologies.

Some approaches solely scrutinise the last snapshot of the generated and original sequences. Notable examples include MTM [14], which employs the Kolmogorov–Smirnov test, and AGE [4] and D2G2 [22], utilising Maximum Mean Discrepancy (MMD), distance on final snapshot distributions. Alternatively, certain methods assume snapshot alignment, allowing paired snapshots for comparison. [23] and [7] deploy 1D metrics and scalar statistics, respectively, with temporal aggregation methods varying from summing absolute differences to using median over absolute differences. Despite these innovations, the evaluation of node-level distributions, as in [2], introduces the oversensitivity of MMD [16]. In contrast, DYMOND [21] presents an approach that avoids assuming snapshot alignment. By computing node-level statistics, specifically the Interquartile Range (IQR) for each node across the time sequence, the authors perform a 2D Kolmogorov-Smirnov test for sequence comparison. Although this method relies on weaker assumptions regarding node ID alignment within a time sequence, its time-invariant nature limits robust temporal comparisons when shuffling time sequences.

Despite these diverse evaluation methodologies, none appear adequately expressive without imposing excessive assumptions. The scrutiny of only the last snapshot mirrors the static graph generation context, overlooking the temporal significance. Snapshot alignment, while a weaker assumption, lacks guarantees regarding sequence alignment, considering variations in length and sampling frequency. Intra-series node alignment, while reasonable, falls short of ensuring time-invariance, which is essential for comprehensive temporal comparisons.

Transitioning from static to temporal context, certain statistics explicitly designed for temporal considerations emerge. Metrics like the number of temporal events, entire network lifespan, mean inter-arrival time, maximum events on an edge, nodes' temporal correlation, and overall temporal correlation coefficient embody a temporal-aware approach. However, challenges persist in their evaluation, including dependency on datasets with attributed nodes, oversight of topology accuracy, and in discrepancies arising from generative use-case parameterisations. While these time-based statistics contribute insights into generation fidelity, they fall short of addressing the full spectrum of limitations inherent in static domain statistics, particularly concerning the temporal evolution of topology.

In evaluating the quality of synthetic data, the practical criterion lies in its utility, gauged by its ability to encapsulate the same valuable information as real data. This equivalence ensures that downstream tasks can perform comparably on both synthetic and real data. In the realm of graph data, particularly in the context of temporal graphs, two crucial tasks for assessing synthetic data utility are link prediction and node labelling. These tasks inherently consider the temporal dimension, acknowledging that the temporal evolution significantly impacts the distributions of links and labels within the graph. Notably, we underscore the integration of utility evaluation with downstream tasks, a facet exemplified by TAGGEN [23] that delves into discussions on anomaly detection and link prediction, ELSM [8] that performs the task of community detection and link prediction to demonstrate the utility and effectiveness of their generative model while Temporal Graph Benchmark (TGB) [9] encompasses dynamic link property prediction and Dynamic Node Property Prediction, providing a comprehensive evaluation scenario for synthetic data utility.

Furthermore, the literature lacks discussions on privacy concerns related to generative models, emphasising the necessity for a solid framework to evaluate privacy. In response to this gap, our work introduces a comprehensive evaluation framework that adapts the graph NNDR metric, as proposed in [11], to the temporal domain.

We underscore the absence of a standardised evaluation framework encompassing topology quality, utility and privacy. Through these contributions, our work addresses critical gaps in the existing literature and establishes a foundation for evaluating the temporal aspects of generative models.

3 Our evaluation framework

3.1 Problem Statement

Graphs are represented by their node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, and by their adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, with N being the maximum number of nodes in the graph, and D being the number of node features. Dynamic graphs are represented as a time series of graphs, $G(t) = (\mathbf{X} \in \mathbb{R}^{T \times N \times D}, \mathbf{A} \in \{0,1\}^{T \times N \times N})$, with T being the number of associated snapshots or timestamps. This snapshot-based representation is sometimes called Discrete-time Dynamic Graphs. A temporal graph series can also be represented in an event-based fashion (Continuous-time Dynamic Graphs), that is, with a set of time-stamped events such as node or edge addition/deletion, or attribute modification. One can change a graph from one representation to the other without loss of information. Event-based representation is usually more space-efficient, but the snapshot-based one is easier to work with, and most models thus use the latter.

Assuming a set of i.i.d. dynamic graphs g $P(X, A)$, the task of temporal graph generation consists of learning an estimate $\hat{P}(X, A)$ of the Distribution $P(X, A)$ of an original dataset, from which new temporal graphs are sampled $\hat{(g)}$ $\hat{P}(X, A)$. However, in many use cases of temporal graphs, such as

social networks, citation networks, or transaction graphs, a single graph holds all the associated data. In general, one cannot guarantee that different sub-graphs or snapshots follow the same distribution. Thus, we are interested in the ability of models to learn patterns from within a single time series and produce new graphs similar to the source one.
As shown in the previous section, the literature still lacks a robust and unified framework to benchmark temporal graph generative models. Some make assumptions that are too strong to provide truly expressive metrics, and some ignore temporality altogether. Most of them don't evaluate the generated node features (if they generate them in the first place) or assume snapshot and node alignment between original and generated sequences. Moreover, no paper we have read tries to assess the privacy or novelty of the obtained generated datasets.
To alleviate all these caveats in the current literature, we propose a new, robust framework to evaluate the performance of a graph generative model accurately. For this, we propose metrics in the following categories: temporal evolution of graph statistics, model-based metrics, and temporal metrics.

3.2 Temporal evolution of graph statistics

In the static context, a very popular way of comparing graphs is to compute a set of topological metrics on both graphs and then compare them with some form of distribution distance, often MMD. In this work, we propose an adaptation of this to the temporal context. This evaluation goes as follows.

1. First, compute statistics (e.g. node degree distribution) on each graph of both sequences (generated and original)
2. Then, for the distribution of each snapshot in each sequence, compute the quantiles of the distribution (the quantiles must be predefined).
3. Finally, compute the multidimensional Dynamic Time-Warping distance (DTW) [19] between the two sequences on the obtained series of quantiles.

As such, the sequences can be of varying length and sampling frequency and the Dynamic Time-Warping algorithm will find the best alignment between snapshots itself; we make no assumption on snapshot alignment. Moreover, by comparing quantiles of the distributions of the snapshots, we get same lengths vectors that can be compared between themselves with an ℓ_1 or ℓ_2 distance without assuming that nodes are aligned between the two sequences. Furthermore, comparing quantiles is useful because it makes no assumption about the underlying distribution of the statistics.

This method is also flexible as it can be used with any statistic, whether it be a distribution or a scalar (in which case, there is no need to compute quantiles), and it can be used with any number of quantiles. More quantiles will give a more sensitive metric at the cost of some computational efficiency.

3.3 Model-based metrics

In this section, we present two metrics that each train a model for a specific task and the resulting metric is the performance of the model on that task. These are more sensitive to the hyperparameters of the chosen model but are here to mimic a real use-case of generated data, i.e. to see if it can be used in a downstream task and to see if models trained on the generated dataset generalise to the original data.

- **Node classification score** Node classification is a frequent task in Graph Machine Learning. The purpose of this metric is to see if an out-of-the-box temporal graph machine learning algorithm is able to classify when it was trained on generated data accurately. Thus the score of this metric is the macro-level ROC-AUC of a temporal node classifier trained on the generated dataset and tested on the original.
- **Link forecasting score** Similarly, link forecasting is one of the main downstream tasks that can be used on generated data. For this metric, the score is the binary ROC-AUC of a temporal link prediction model trained on the generated dataset and tested on the original.

3.4 Temporal metrics

We also include intrinsically temporal metrics directly from the state of the art such as temporal correlation[22]. Time measurement metrics, such as mean inter-arrival time[14], might not correctly evaluate generation on some use cases (e.g. with a dilated time dimension) so we removed them from our evaluation.

3.5 Privacy assessment

In our research, we're breaking new ground by assessing the privacy of time-varying graphs generated by models. This unique approach is the first of its kind, as it focuses on the privacy aspects of these evolving graphs.
Following [11] evaluation framework, we use the Nearest Neighbour Distance Ratio (NNDR) method. The main idea behind this metric is to train an embedder on the original data for a downstream task and then measure the distances between the embeddings of nodes in the original graph and those in a generated graph. However, the metric needs to account for nodes appearing and disappearing over time, and transforming. This dynamic nature makes it tricky to figure out who the "nearest neighbours" are, and it complicates the calculations. To handle this, we introduce temporal NNDR. In this approach, we work with graph snapshots. We assume that the nodes of these snapshots are aligned. More precisely, if a node disappears, it becomes isolated, and if a new node appears, it's considered as if it used to be an isolated node. This creates a bound over the number of nodes considered in the graph; this representation solves the issue of arbitrary length representations of the graph, at the cost of a restricted evaluation method that would need to be updated whenever

the graph series were to grow past that bound.

The NNDR metric can be extended to the temporal setting in two ways. Considering the following method of obtaining the NNDR metric:

1. First train an embedder (temporal link predictor or node classifier) on original dataset.
2. Once trained, this embedder is used to get embedding from original and generated sets.
3. Then we calculate a distance matrix between these embeddings
4. Finally, derive NNDR from this matrix to assess privacy.

The first way to account for time is to use a temporal-aware distance to generate the distance matrix, such as DTW [19]. In that case, the embeddings can be generated for each node at each snapshot.
The second way uses temporal-aware embeddings instead; keeping regular l_1 or l_2 distances for the distance matrix. A recursive model can be used over each subsequent snapshot of the temporal graph. The model's hidden state should propagate useful information from the previous snapshots to the next, thus, the embeddings of the final snapshot are used to represent the whole temporal graph's embeddings. While we are considering only the embedding of the last snapshot, a recusive model can embed the whole series. Such an embedder avoids the issue of the distribution comparison of last snapshot topology statistics discussed in 2.

4 Experiments

4.1 Generative models description

Our evaluation is limited to models in temporal graph generation that were successfully executed. The subsequent table enumerates these models, capturing fundamental aspects of their methodologies. We highlight whether a model captures the probability distribution of the entire graph ($P(G)$) or models temporal dependencies explicitly ($P(G_t|G_{t-1})$), illustrating their approaches in representing evolving graph structures. Additionally, the table delves into the strategies employed for graph generation, such as random walk-based[1] and motif-based[2] approaches. It also flags models leveraging node and edge features, emphasizing their capability to incorporate nodal attributes and relationships between nodes in the temporal graph generation process. This comprehensive table serves as a valuable reference, providing insights into the distinctions among SOTA models concerning critical aspects.

[1]The generator simulates the process of nodes navigating the graph through random walks, determining the sequence of node visits based on probabilistic decisions at each step.
[2]The generator identifies and replicates motifs, contributing to the formation of the overall graph structure.

Model	Temporal modelling	Random walk-based	Motif based	Node features	Edge features
DYMOND [21]	$P(G)$	X	✓	X	X
DAMNETS [2]	$P(G_t\|G_{t-1})$	X	X	X	X
TIGGER [7]	$P(G)$	✓	X	X	X
D2G2 [22]	$P(G)$	X	X	✓	X
AGE [4]	$P(G_t\|G_{t-1})$	X	X	✓	X

4.2 Data Description

We utilise two featured datasets from [18]. The TwitterTennis dataset portrays Twitter accounts as nodes, with edges denoting mentions. EnglandCovid, sourced from Facebook Data For Good disease prevention maps, tracks daily movements between regions. Additional non-featured datasets from [7] include the Bitcoin alpha network, a homogeneous financial transaction graph capturing bitcoin trading, and the Reddit Interaction network, a bipartite graph illustrating user interactions on subreddits. Introducing Wiki-Small, a bipartite graph capturing initial Wikipedia edits over 50 hours, and Insecta [17], a temporal graph capturing ant colony dynamics every half-second over 41 days. Finally, the IMDB Dynamic dataset [17] documents director-actor collaborations in movies from 1980 to 2007. Table 1 summarises the used dataset characteristics (number of time snapshots, number of nodes, minimum number of nodes, minimum, mean and maximum number of edges over time, and the dimension of attribute data in nodes)

4.3 Experimental protocol

All experiments are performed on a machine running Intel(R) Xeon(R) Gold 6134 CPU@3.20GHz processor with 32 physical cores, with 1 Nvidia Tesla V100 card with 32GB GPU memory, and 128GB RAM with Ubuntu 18.04.6 operating system.
We train all models from the SOTA using automated parameter optimization (optuna) over all datasets. We then generate synthetic graphs for each dataset/model pair. The synthetic graphs are generated to contain the same amount of timesteps than the original graph dataset.
For auto-regressive models, since they generate each snapshot based on the previous snapshot, 2 generation methods are possible: either the model generates the whole sequence based only on a single, first snapshot from the original data (full series generation), or the real data is used to generate each snapshot sequentially (1 snapshot ahead generation).
Every node without any edges in any snapshot is also removed from the graphs. Also, Even though the AGE model is supposed to generate node features, we were unable to run an implementation that actually did.

4.4 Results and analysis

The metrics used were the following : Spectral, degree, degree centrality, local cluster coefficient, closeness centrality,

Table 1. Temporal graph datasets characteristics

	n. snapshots	*n. nodes*	*min n. edges*	*mean n. edges*	*max n. edges*	*Node features*
Insecta	39	165	4894	12677.64	21548	X
Twitter Tennis	120	1000	41	340.32	936	16
England Covid	53	129	836	1358.5	2158	8
Wiki small	50	1616	33	59.60	93	X
Bitcoin	190	3783	1	127.29	1171	X
Reddit	588915	10984	1	1.14	7	X
IMDB	28	150544	6822	21156.29	37040	X

eigenvalue centrality, average cluster coefficient and finally, transitivity are all statistical-based metrics temporally aggregated using the technique in section 3.2. As a utility metric, we used link prediction task, evaluated with AUROC value. Node labelling was left out due to most datasets not having node labels. Finally, temporal correlation, temporal closeness and temporal clustering coefficient were used as time-based statistical metrics.

Table 2 show results on the England Covid dataset. DYMOND outshines D2G2 across various aspects, exhibiting superior results in spectral analysis, node degree, and closeness centrality. Notably, D2G2 faces challenges in spectral and temporal correlation metrics, indicative of potential limitations in capturing intricate temporal dependencies. On the contrary, DYMOND has better performance in link prediction AUC, demonstrating its proficiency in preserving network utility.

Table 2. Quality metrics of generative models on England Covid Dataset

	Generative models	
	D2G2	DYMOND
↘ *spectral*	0.015	**0.011**
↘ *degree*	0.018	**0.009**
↘ *deg. cent.*	**0.000**	0.001
↘ *local clust. coeff.*	**0.019**	0.025
↘ *close. cent.*	**0.002**	0.004
↘ *eigen. cent.*	**0.018**	0.029
↘ *ave. clust. coeff.*	**0.731**	1.814
↘ *transitivty*	**0.211**	0.797
↗ *link pred. AUC*	0.564	**0.656**
↘ *temp. corr.*	**86098.067**	86098.337
↘ *temp. close. diff.*	17.427	**9.097**
↘ *temp. clust. coeff. diff.*	**0.031**	0.405

Table 3. Quality metrics for update models on England Covid dataset

	1 snapshot ahead		*Full series*	
	AGE	DAMNETS	AGE	DAMNETS
↘ *spectral*	0.004	**0.004**	0.014	**0.009**
↘ *degree*	**0.003**	0.004	0.011	**0.006**
↘ *deg. cent.*	**0.000**	**0.000**	**0.000**	**0.000**
↘ *local clust. coeff.*	0.006	**0.005**	0.026	**0.018**
↘ *close. cent.*	0.002	**0.001**	0.003	**0.003**
↘ *eigen. cent.*	0.010	**0.007**	0.035	**0.027**
↘ *ave. clust. coeff.*	0.852	**0.415**	**0.797**	1.556
↘ *transitivty*	0.482	**0.216**	**0.237**	0.881
↗ *link pred. AUC*	**0.604**	0.590	0.541	**0.624**
↘ *temp. corr.*	86097.977	**86097.911**	**86097.781**	86097.791
↘ *temp. close. diff.*	**15.783**	31.457	**14.868**	30.484
↘ *temp. clust. coeff. diff.*	0.392	**0.283**	0.281	**0.249**

As for update models, AGE and DAMNETS, DAMNETS consistently outperforms AGE across multiple metrics, showcasing its strengths in spectral analysis, node degree, and local clustering coefficient. Notably, both models exhibit comparable results in degree centrality as mentionned in table A, emphasizing their effectiveness in preserving node connectivity. However, DAMNETS excels in capturing local clustering patterns, as evidenced by its lower values in the respective metrics. Additionally, DAMNETS demonstrates superior performance in link prediction AUC for the full series, highlighting its efficiency in preserving network utility over time.

More evaluation results are available in Appendix Section A.

Privacy sensitivity analysis: Table 4 shows a sensitivity analysis run on England Covid and Twitter Tennis datasets, each considered with and without node features. We performed this study to validate the capability of temporal NNDR to follow privacy levels. Overall, NNDR scores exhibit a gradual increase in response to perturbations on datasets with node features, while being stable in their absence. This means the metric is apt at detecting privacy leaks in attribute data, but is less reliable to measure privacy in the topology. Temporal NNDR is a robust privacy metric, but is limited in applicability; this shows the need for further contribution in privacy

assessment on graph generation.

Table 4. Sensitivity analysis of the NNDR (mean ± stand. dev.) metric with DTW method.

Datasets	England Covid		Twitter Tennis	
	w.o. n.f	*w. n.f*	*w.o. n.f*	*w. n.f*
Orig. data	**0 ± 0**	**0 ± 0**	0.56 ± 0.48	**0.01 ± 0.10**
Pert. data (5%)	0.89 ± 0.12	**0.32 ± 0.1**	0.89 ± 0.12	**0.59 ± 0.19**
Pert. data (10%)	0.91 ± 0.10	**0.40 ± 0.13**	0.90 ± 0.11	**0.63 ± 0.17**
Pert. data (25%)	0.92 ± 0.09	**0.54 ± 0.17**	0.91 ± 0.09	**0.71 ± 0.14**
Pert. data (50%)	0.93 ± 0.06	**0.69 ± 0.18**	0.93 ± 0.07	**0.79 ± 0.12**
Pert. data (75%)	0.92 ± 0.07	**0.75 ± 0.16**	0.93 ± 0.06	**0.83 ± 0.10**
Pert. data (90%)	0.93 ± 0.06	**0.78 ± 0.15**	0.94 ± 0.06	**0.85 ± 0.1**

Privacy score calculation methods: The NNDR calculation methods, employing Dynamic Time Warping (DTW) on the full embeddings matrix over snapshots, l_1 norm, and l_2 norm on the last snapshot embeddings, provide insightful observations on the privacy assessment on the generated temporal graphs, as illustrated in Tables 5 and 6. These tables present the mean and standard deviation of NNDR scores across different perturbation percentages for the EnglandCovid and TwitterTennis datasets, respectively. Perturbation at a rate of $p\%$ involves modifying the original data by replacing $p\%$ of the edges with random ones and altering $p\%$ of the node feature matrix.

Table 5 shows the NNDR scores for the EnglandCovid dataset. The scores remain consistently low across perturbation percentages for all three methods. This tendency indicates that the embedder, trained on the original data and evaluated using the NNDR metric on the perturbed ones, maintains a robust ability to capture the privacy aspects of evolving graphs. The slight fluctuations in scores can be attributed to the varying impact of perturbations on the temporal dynamics of the graph.

Table 6, the NNDR scores for the TwitterTennis dataset reveal interesting trends. As the perturbation percentage increases, there is a general increase in NNDR mean values across all three calculation methods. This upward trend suggests that the privacy aspects of the temporal graph become more challenging to preserve as perturbations intensify. The variations in NNDR scores highlight the sensitivity of the metric to changes in the node features and the effectiveness of the embedder in capturing these variations.

The robustness and validity of employing Dynamic Time Warping (DTW), l_1 norm, and l_2 norm in the Nearest Neighbor Distance Ratio (NNDR) calculation methods are underpinned by their capacity to account for the evolving nature of time-varying graphs. The choice between using the DTW calculation method on the full embedding matrix or the l_1 or l_2 norms on the last embeddings only introduces flexibility in handling the dynamic nature of the graph series. The decision to use DTW accounts for temporal misalignments, generating distance matrices that reflect the temporal aspects of the graph. Conversely, the use of temporal-aware embeddings incorporates the information from previous snapshots through a recursive model, providing a different perspective on the evolving graph's privacy. The experimental results underscore the validity of these calculation methods. Moreover, the trends observed in both the EnglandCovid and TwitterTennis datasets show that the NNDR scores increase with higher perturbation percentages, highlighting the sensitivity of the metric to changes in node features and topological structure. This aligns with the expectation that, as perturbations intensify, preserving privacy becomes more challenging.

Privacy score on generated graphs: The table in 7 showcases Nearest Neighbor Distance Ratio (NNDR) scores for the EnglandCovid and Insecta datasets. As perturbation levels increase, there is a corresponding rise in NNDR scores, indicating the impact on privacy. Notably, the D2G2 model exhibits superior performance, achieving remarkable NNDR scores (0.982 ± 0.079 for EnglandCovid and 0.99 ± 0.001 for Insecta) on par with significant perturbation (90%). This underscores the model's robustness in preserving privacy across both datasets.

Moreover, a sensitivity analysis is undertaken on the unfeatured Insecta dataset. This study involves incorporating node degree, clustering coefficient, and spectral features into the evaluation's feature matrix. The detailed results are provided in Appendix Section B.

5 Conclusion

We have evaluated state-of-the-art temporal graph generative models, recognizing the significance of a unified benchmark across various datasets. Our proposed rigorous evaluation protocol addresses the crucial aspects of temporal generation quality, utility, and privacy, providing a comprehensive measure of model performance. The success of a temporal graph generative model hinges on three fundamental pillars. Firstly, scalability to large temporal graphs is crucial, considering the expansive nature of real-world graph structures. While our paper extensively covers the aspects of generation quality and utility, we acknowledge that further work is needed to explicitly address scalability in our evaluation protocol. Secondly, the model's proficiency in accurately learning the underlying distribution ensures fidelity and utility for downstream tasks. Lastly, the preservation of privacy is paramount, especially in applications involving sensitive information. **Future Work:** Expanding the scope to include more downstream tasks could enhance the applicability of generative temporal graph models. Additionally, investigating novel approaches to further improve scalability, distribution learning, and privacy preservation would contribute to advancing the state of the art in this

Table 5. NNDR score on perturbed England Covid dataset with node features using DTW, l_1, l_2

	DTW	l_1	l_2
Perturbation Percentage	*NNDR (mean ± std)*	*NNDR (mean ± std)*	*NNDR (mean ± std)*
Orig. data	0 ± 0	0 ± 0	0 ± 0
Pert. data (5%)	0.323 ± 0.1	0.435 ± 0.244	0.450 ± 0.228
Pert. data (10%)	0.397 ± 0.127	0.527 ± 0.247	0.536 ± 0.249
Pert. data (25%)	0.543 ± 0.166	0.652 ± 0.231	0.695 ± 0.230
Pert. data (50%)	0.694 ± 0.179	0.768 ± 0.194	0.760 ± 0.193
Pert. data (75%)	0.75 ± 0.162	0.815 ± 0.156	0.800 ± 0.187
Pert. data (90%)	0.784 ± 0.155	0.820 ± 0.148	0.821 ± 0.159

Table 6. NNDR score on perturbed Twitter Tennis dataset with node features using DTW, l_1, l_2

	DTW	l_1	l_2
Perturbation Percentage	*NNDR (mean ± std)*	*NNDR (mean ± std)*	*NNDR (mean ± std)*
Orig. data	0.01 ± 0.1	0.016 ± 0.126	0.016 ± 0.126
Pert. data (5%)	0.589 ± 0.188	0.498 ± 0.341	0.492 ± 0.338
Pert. data (10%)	0.631 ± 0.173	0.485 ± 0.339	0.495 ± 0.340
Pert. data (25%)	0.709 ± 0.145	0.509 ± 0.330	0.512 ± 0.328
Pert. data (50%)	0.788 ± 0.117	0.547 ± 0.323	0.558 ± 0.313
Pert. data (75%)	0.83 ± 0.1	0.590 ± 0.300	0.558 ± 0.304
Pert. data (90%)	0.848 ± 0.097	0.590 ± 0.293	0.595 ± 0.298

Table 7. NNDR (mean ± stand. dev.) of generated graphs with DTW method.

	Datasets	
	EnglandCovid	Insecta
Orig. data	0 ± 0	0 ± 0
Pert. data (50%)	0.694 ± 0.179	0.886 ± 0.112
Pert. data (75%)	0.75 ± 0.162	0.901 ± 0.107
Pert. data (90%)	0.784 ± 0.155	0.912 ± 0.106
D2G2	**0.982 ± 0.079**	**0.990 ± 0.001**

nascent field. Furthermore, exploring the adaptation of these models to domain-specific challenges and datasets may yield valuable insights and applications. Finally, our evaluation framework currently has minimal evaluation in privacy and scalability; the evaluation for both of them has a lot of room for improvement.

References

[1] Xiaohui Chen, Yukun Li, Aonan Zhang, and Li-ping Liu. 2022. NVDiff: Graph Generation through the Diffusion of Node Vectors. *arXiv preprint arXiv:2211.10794* (2022).

[2] Jase Clarkson, Mihai Cucuringu, Andrew Elliott, and Gesine Reinert. 2022. DAMNETS: A deep autoregressive model for generating Markovian network time series. In *Learning on Graphs Conference*. PMLR, 23–1.

[3] Hanjun Dai, Azade Nazi, Yujia Li, Bo Dai, and Dale Schuurmans. 2020. Scalable deep generative modeling for sparse graphs. In *International conference on machine learning*. PMLR, 2302–2312.

[4] Shuangfei Fan and Bert Huang. 2020. Attention-based graph evolution. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24*. Springer, 436–447.

[5] Nikhil Goyal, Harsh Vardhan Jain, and Sayan Ranu. 2020. GraphGen: a scalable approach to domain-agnostic labeled graph generation. In *Proceedings of The Web Conference 2020*. 1253–1263.

[6] Dezhi Guo, Zhaowei Liu, and Ranran Li. 2023. RegraphGAN: A graph generative adversarial network model for dynamic network anomaly detection. *Neural Networks* 166 (2023), 273–285.

[7] Shubham Gupta, Sahil Manchanda, Srikanta Bedathur, and Sayan Ranu. 2022. TIGGER: Scalable Generative Modelling for Temporal Interaction Graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 6819–6828.

[8] Shubham Gupta, Gaurav Sharma, and Ambedkar Dukkipati. 2018. A Generative Model for Dynamic Networks with Applications. arXiv:1802.03725 [cs.SI]

[9] Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. 2023. Temporal Graph Benchmark for Machine Learning on Temporal Graphs. arXiv:2307.01026 [cs.LG]

[10] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. 2022. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*. PMLR, 10362–10383.

[11] Valentin Lemaire, Youssef Achenchabe, Lucas Ody, Houssem Eddine Souid, Gianmarco Aversano, Nicolas Posocco, and Sabri Skhiri. 2023. SANGEA: Scalable and Attributed Network Generation.

arXiv:2309.15648 [cs.LG]
[12] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Charlie Nash, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard S. Zemel. 2019. Efficient Graph Generation with Graph Recurrent Attention Networks. *CoRR* abs/1910.00760 (2019).
[13] Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. 2019. Graph Normalizing Flows. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.
[14] Penghang Liu and Ahmet Erdem Sariyüce. 2023. Using Motif Transitions for Temporal Graph Generation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1501–1511.
[15] Maria Massri, Zoltan Miklos, Philippe Raipin, Pierre Meye, Amaury Bouchra Pilet, and Thomas Hassan. 2023. RTGEN++: A relative temporal graph generator. *Future Generation Computer Systems* 146 (2023), 139–155.
[16] Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. 2021. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. *arXiv preprint arXiv:2106.01098* (2021).
[17] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. https://networkrepository.com
[18] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, , Guzman Lopez, Nicolas Collignon, and Rik Sarkar. 2021. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. 4564–4573.
[19] Mohammad Shokoohi-Yekta, Bing Hu, Hongxia Jin, Jun Wang, and Eamonn J. Keogh. 2016. Generalizing DTW to the multi-dimensional case requires an adaptive approach. *Data Mining and Knowledge Discovery* 31 (2016), 1–31. https://api.semanticscholar.org/CorpusID:26125592
[20] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. GraphRNN: A Deep Generative Model for Graphs. *CoRR* abs/1802.08773 (2018).
[21] Giselle Zeno, Timothy La Fond, and Jennifer Neville. 2021. Dymond: Dynamic motif-nodes network generative model. In *Proceedings of the Web Conference 2021*. 718–729.
[22] W Zhang, LM Zhang, D Pfoser, and L Zhao. [n. d.]. Disentangled Dynamic Graph Deep Generation. arXiv 2021. *arXiv preprint arXiv:2010.07276* ([n. d.]).
[23] Dawei Zhou, Lecheng Zheng, Jiawei Han, and Jingrui He. 2020. A data-driven graph generative model for temporal interaction networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 401–411.

A Quality Metrics

The appendices contain extra results, over more datasets and models. In evaluating the generative models in table 8, TIGGER, D2G2, and DYMOND on the Insecta Dataset, a nuanced perspective emerges. TIGGER exhibits exceptional performance in topological quality and link prediction that makes of the generated graph more faithful to the original data.

In table 9, DAMNETS demonstrates superiority, notably excelling in topological quality and link prediction AUC. Despite AGE's competitive edge in certain topological aspects,

Table 8. Quality metrics of generative models on Insecta Dataset

	Generative models		
	TIGGER	D2G2	DYMOND
↘ *spectral*	4.470	7.829	**3.498**
↘ *degree*	**473.203**	796.875	743.752
↘ *deg. cent.*	804.748	**0.000**	**0.000**
↘ *local clust. coeff.*	**3.855**	6.146	5.611
↘ *close. cent.*	3.895	5.257	**2.965**
↘ *eigen. cent.*	0.341	0.358	**0.193**
↘ *ave. clust. coeff.*	**2.453**	4.077	3.539
↘ *transitivty*	**1.301**	4.885	4.433
↗ *link pred. AUC*	**0.894**	0.479	0.555
↘ *temp. corr.*	10.193	**10.113**	10.714
↘ *temp. close. diff.*	**0.287**	161.167	122.511
↘ *temp. clust. coeff. diff.*	**0.108**	0.479	0.463

Table 9. Quality metrics for update models on Insecta dataset

	1 snapshot ahead		*Full series*	
	AGE	DAMNETS	AGE	DAMNETS
↘ *spectral*	3.835	**3.818**	3.999	**3.996**
↘ *degree*	**87.221**	127.284	349.574	**346.909**
↘ *deg. cent.*	**4.937**	**4.937**	**4.937**	**4.937**
↘ *local clust. coeff.*	**1.670**	2.444	**2.729**	2.928
↘ *close. cent.*	2.227	**1.961**	2.906	**2.863**
↘ *eigen. cent.*	**0.076**	0.092	0.331	**0.323**
↘ *ave. clust. coeff.*	**0.129**	0.457	**0.290**	0.600
↘ *transitivty*	**0.555**	0.999	**1.186**	1.484
↘ *diameter*	**2.236**	2.646	2.828	**2.236**
↘ *ave. short. path length*	1.224	**0.855**	**0.500**	1.090
↗ *link pred. AUC*	0.570	**0.770**	**0.559**	0.548
↘ *temp. corr.*	**10.166**	10.231	**9.870**	10.001
↘ *temp. close. diff.*	**80.354**	80.404	80.612	**80.387**
↘ *temp. clust. coeff. diff.*	**0.248**	0.295	**0.210**	0.253

DAMNETS emerges as the more well-rounded model, showcasing enhanced utility in preserving link structures. Both models exhibit comparable temporal characteristics and privacy preservation, but DAMNETS stands out as the preferred choice.

The generative models' performance in table 10 on the Wiki-small dataset reveals distinct characteristics. D2G2 faces a significant scaling challenge, as indicated by an Out of Memory (OOM). DYMOND and TIGGER, on the other hand, show good topological fidelity, but very poor utility, as indicate by the Link prediction task, indicating a strong overfitting risk.

The table 11 presents performance metrics for the generative models AGE and DAMNETS in both one-snapshot ahead and

Table 10. Quality metrics for Generative models on Wiki-small dataset

	Generative models		
	DYMOND	D2G2	TIGGER
↘ *spectral*	**0.000**		**0.000**
↘ *degree*	**0.000**		**0.000**
↘ *deg. cent.*	**0.000**		8966.633
↘ *local clust. coeff.*	**0.000**		**0.000**
↘ *close. cent.*	**0.000**		**0.000**
↘ *katz cent.*	**0.001**		0.071
↘ *eigen. cent.*	0.000	OOM	**0.000**
↘ *ave. clust. coeff.*	**0.000**		0.002
↘ *transitivty*	**0.000**		0.140
↗ *link pred. AUC*	**0.502**		**0.502**
↘ *temp. corr.*	0.194		**0.038**
↘ *temp. close. diff.*	**0.111**		10.524
↘ *temp. clust. coeff. diff.*	**0.000**		0.001

Table 11. Quality metrics for update models on Wiki Small dataset

	1 snapshot ahead		*Full series*	
	AGE	DAMNETS	AGE	DAMNETS
↘ *spectral*	10.083	**0.000**	9.923	**8.948**
↘ *degree*	180.380	**0.000**	3389.401	**29.139**
↘ *deg. cent.*	**0.000**	5.590	**0.000**	491.935
↘ *local clust. coeff.*	0.158	**0.000**	2.164	**0.000**
↘ *close. cent.*	3.496	**0.000**	5.440	**1.539**
↘ *katz cent.*	**0.001**	0.003	**0.001**	0.071
↘ *eigen. cent.*	0.182	**0.000**	0.239	**0.052**
↘ *ave. clust. coeff.*	0.105	**0.000**	1.533	**0.008**
↘ *transitivty*	0.138	**0.000**	1.554	**0.006**
↗ *link pred. AUC*	0.499	**0.501**	0.498	**0.500**
↘ *temp. corr.*	0.216	**0.135**	0.707	**0.703**
↘ *temp. close. diff.*	543.125	**4.415**	563.458	**2.308**
↘ *temp. clust. coeff. diff.*	0.019	**0.000**	0.142	**0.000**

full-series scenarios. DAMNETS consistently outperforms AGE across key measures, including spectral analysis, degree, degree centrality, local clustering coefficient, and closeness centrality, showcasing its superior ability to capture network structure.

The table 12 reveals the performance metrics of generative models, D2G2 and TIGGER, on the Bitcoin dataset. D2G2 encounters an Out of Memory (OOM) issue, mirroring its challenges seen in a similar scenario with the Wiki-small dataset. In addition, TIGGER demonstrates an alarming degree of overfitting. While achieving a perfect link prediction

Table 12. Quality metrics for Generative models on Bitcoin dataset

	Generative models	
	D2G2	TIGGER
↘ *spectral*		**0.000**
↘ *degree*		**0.000**
↘ *deg. cent.*		**0.002**
↘ *local clust. coeff.*		**0.000**
↘ *close. cent.*		**0.000**
↘ *katz cent.*		**0.002**
↘ *eigen. cent.*	OOM	**0.000**
↘ *ave. clust. coeff.*		**0.005**
↘ *transitivty*		**0.566**
↗ *link pred. AUC*		**0.883**
↘ *temp. corr.*		**0.212**
↘ *temp. close. diff.*		**32.103**
↘ *temp. clust. coeff. diff.*		**0.001**

AUC score of 0.883 may seem promising on the surface, such perfection raises suspicions of over-optimization and an inability to generalize well to unseen data. The model's exceptionally low topological metrics difference (0.001) shows a risk of overfitting.

Table 13. Quality metrics for update models on Twitter Tennis dataset

	1 snapshot ahead	*Full series*
	AGE	AGE
↘ *spectral*	**16.045**	**15.471**
↘ *degree*	**135.481**	**4550.644**
↘ *deg. cent.*	**0.000**	**0.000**
↘ *local clust. coeff.*	**0.163**	**4.638**
↘ *close. cent.*	**4.503**	**8.928**
↘ *katz cent.*	**0.023**	**0.023**
↘ *eigen. cent.*	**0.371**	**0.482**
↘ *ave. clust. coeff.*	**0.057**	**3.150**
↘ *transitivty*	**1.896**	**1.902**
↗ *link pred. AUC*	**0.586**	**0.706**
↘ *temp. corr.*	**0.099**	**0.853**
↘ *temp. close. diff.*	**342.871**	**352.025**
↘ *temp. clust. coeff. diff.*	**0.004**	**0.155**

Similar to observations in previous datasets, the table 14 reveals how D2G2 encounters an Out of Memory (OOM) issue during spectral analysis, indicating potential memory

Table 14. Quality metrics for Generative models on Twitter Tennis dataset

	Generative models	
	D2G2	TIGGER
↘ *spectral*		**0.015**
↘ *degree*		**0.008**
↘ *deg. cent.*		**0.934**
↘ *local clust. coeff.*		**0.000**
↘ *close. cent.*		**0.016**
↘ *katz cent.*		**0.003**
↘ *eigen. cent.*	OOM	**0.012**
↘ *ave. clust. coeff.*		**0.299**
↘ *transitivty*		**1.802**
↗ *link pred. AUC*		**0.757**
↘ *temp. corr.*		**0.116**
↘ *temp. close. diff.*		**335.115**
↘ *temp. clust. coeff. diff.*		**0.035**

constraints. This recurring challenge raises concerns about D2G2's adaptability to diverse network structures, echoing its limitations seen in other datasets. Furthermore, although TIGGER successfully mitigates out-of-memory challenges, its susceptibility to overfitting is evident through significantly low values in metrics such as spectral, degree and average clustering coefficients. This tendency towards overfitting is particularly noteworthy considering the distinctive temporal dynamics shared between the Twitter Tennis dataset and previously examined datasets like wiki-small and Bitcoin, which exhibit similar network properties. Addressing this challenge requires a nuanced approach, emphasizing model adjustments or the incorporation of regularization techniques to enhance the generalization capabilities of TIGGER.

B Privacy

Focusing on the England Covid and Insecta datasets, we systematically compared the performance of the original feature set ("orig n.f") against a modified feature set that incorporates spectral, clustering, and node degree features ("w. deg. spec. clust. f"). Notably, as perturbation increased, both the original and modified feature sets demonstrated a proportional rise in NNDR scores. We also considertThe unfeatured version of the Insecta dataset, characterised by an initial baseline privacy level of 0.557 ± 0.497, presented a perplexing stability in metrics ranging from 0.88 to 0.92 across perturbation levels (5% to 90%). This inexplicable constancy contrasts with the expected trend. Intriguingly, the incorporation of additional features consistently yielded lower NNDR scores under perturbation, providing compelling evidence to support the hypothesis that augmenting the feature matrix with spectral, clustering, and node degree features enhances the capabilities of the embedder to learn the graph distribution. This recognition positions these features as valuable components for evaluating D2G2 privacy on the Insecta dataset.

Table 15. Sensitivity analysis of the NNDR (mean ± stand. dev.) metric adding degree, spectral and clustering as features with DTW method.

Datasets	England Covid		Insecta	
	orig n.f	*w. deg. spec. clust .f*	*w.o. n.f*	*w. deg. spec. clust. f*
Orig. data	0 ± 0	0 ± 0	0.56 ± 0.5	**0.01 ± 0.10**
Pert. data (5%)	0.32 ± 0.10	0.46 ± 0.14	0.89 ± 0.13	**0.70 ± 0.21**
Pert. data (10%)	0.4 ± 0.13	0.65 ± 0.16	0.93 ± 0.08	**0.76 ± 0.19**
Pert. data (25%)	0.54 ± 0.17	0.83 ± 0.15	0.93 ± 0.07	**0.84 ± 0.15**
Pert. data (50%)	0.69 ± 0.18	0.90 ± 0.09	0.92 ± 0.07	**0.89 ± 0.12**
Pert. data (75%)	0.75 ± 0.16	0.91 ± 0.07	0.92 ± 0.07	**0.90 ± 0.11**
Pert. data (90%)	0.78 ± 0.15	0.92 ± 0.07	0.92 ± 0.07	**0.91 ± 0.11**