RETHINKING EVALUATION FOR TEMPORAL LINK PREDICTION THROUGH COUNTERFACTUAL ANALYSIS

Aniq Ur Rahman[1] **Alexander Modell**[2] **Justin P. Coon**[1]
[1]Department of Engineering Science, University of Oxford, U.K.
{aniq.rahman, justin.coon}@eng.ox.ac.uk
[2]Department of Mathematics, Imperial College London, U.K.
a.modell@imperial.ac.uk

ABSTRACT

In response to critiques of existing evaluation methods for temporal link prediction (TLP) models, we propose a novel approach to verify if these models truly capture temporal patterns in the data. Our method involves a sanity check formulated as a counterfactual question: "What if a TLP model is tested on a temporally distorted version of the data instead of the real data?" Ideally, a TLP model that effectively learns temporal patterns should perform worse on temporally distorted data compared to real data. We analyse this hypothesis and introduce two temporal distortion techniques to assess six well-known TLP models.

1 INTRODUCTION

In static graphs, link prediction refers to the task of predicting whether an edge exists between two nodes after having observed other edges in the graph. Temporal link prediction (TLP) is a dynamic extension of link prediction wherein the task is to predict whether a link (edge) exists between any two nodes in the future based on the historical observations (Qin and Yeung, 2023). The predictive capability of TLP models make them useful in applications pertaining to dynamic graphs, such as product recommendations (Qin et al., 2024; Fan et al., 2021), social network content or account recommendation (Fan et al., 2019; Daud et al., 2020), fraud detection in financial networks (Kim et al., 2024), and resource allocation, to name a few.

In the TLP literature (Kumar et al., 2019; Trivedi et al., 2019; Xu et al., 2020; Rossi et al., 2020; Wang et al., 2020; Cong et al., 2023; Yu et al., 2023), the TLP task is treated as a binary classification problem where the query

q_1 : "Does an edge exist between the nodes u and v at time t?"

is processed by a model and then compared with the ground truth following which metrics such as area under the receiver operating characteristic curve (AU-ROC), and average precision (AP) are reported. The ground truth consists of positive samples, and a fixed number of random negative samples. There are a couple of issues in the binary classification approach. Firstly, the timestamps in the query are restricted to the timestamps present in the ground truth, which makes the evaluation biased and does not test the model's performance in the continuous time range. Secondly, checking for the existence of an edge at a specific timestamp is an ill-posed question, and instead the existence of an edge should be queried within a finite time-interval. Lastly, the negative edge sampling strategy, and the number of negative samples per positive sample impact the performance metrics as seen in EXH (Poursafaei and Rabbany, 2023).

Alternatively, in a rank-based approach, the query is formulated as:

q_2 : "Which nodes are likely to have an edge with node u at time t?"

In this case, the model returns an ordered list of nodes arranged from most likely to least likely. Then, the rank of the ground truth edge is returned if a match is found, and if not, a high number is reported. For all the edges in the test data, metrics such as Mean Average Rank (MAR) or Mean Reciprocal Rank (MRR) can be reported to assess the performance of the model (Huang et al., 2024). While the rank-based metrics are more intuitive than AU-ROC and AP, the issues regarding binary classification mentioned above still remain unaddressed. To give a true picture of the predictive power

of the TLP models, a penalty term should be introduced to account for the nodes that are incorrectly estimated to form an edge with node u at time t.

In a recent work, Poursafaei et al. (2022) highlighted that the state-of-the-art (SoTA) performance of some TLP models on the standard benchmark datasets is near-perfect. This is counterintuitive because TLP is a challenging task, even more challenging than link prediction of static graphs, due to the additional degree of freedom in the data induced by the temporal dimension. The flaw in the evaluation method is attributed to the limited negative sampling strategy, and the authors propose a new negative edge sampling strategy which results in a different ranking of the baselines.

Inspired by the critique of the evaluation method, we propose a method to conduct sanity check of the TLP models to determine if they truly capture the temporal patterns in the data. The sanity check is formulated as the counterfactual question (Pearl, 2009):

> "What if a TLP model which is trained on a temporal graph is tested on *temporally distorted* version of the data instead of the real data?"

Ideally, a TLP model which is capable of learning the temporal patterns should perform worse on temporally distorted data compared to the real data. We conduct an in-depth analysis of this argument and introduce various data distortion techniques to assess well-known TLP models.

2 Counterfactual Analysis

Definition 2.1. A **temporal graph** with $m \in \mathbb{N}$ instantaneous edges formed between nodes in $\mathcal{U}$ and $\mathcal{V}$ is defined as $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, where $\mathcal{E} \triangleq \{(u_i, v_i, t_i) : i \in [m], u_i \in \mathcal{U}, v_i \in \mathcal{V}, t_i \in \mathbb{R}\}$ denotes the set of edges. The triple (u, v, t) is referred to as an edge event.

Definition 2.2. The slice of edges in $\mathcal{E}$ with timestamps in the range (t_1, t_2) is denoted as $\mathcal{E}(t_1, t_2)$, and defined as $\mathcal{E}(t_1, t_2) \triangleq \{(u, v, t) : (u, v, t) \in \mathcal{E}, t \in (t_1, t_2)\}$.

A temporal graph is characterized by (1) the *order* in which the edges appear, (2) the *frequency* with which edges appear over time, and (3) the *time gap* between any two edge events. In this work, we refer to these characteristics as **temporal patterns**. Furthermore, if temporal patterns observed in the past enable predictions of future temporal patterns that outperform naïve estimates on a specific performance metric, then the temporal data is considered **learnable**. This does not require the temporal pattern to remain consistent over time; rather, it suggests that future changes can be estimated from past observations.

Experiment Setup A model f is trained on a temporal graph $\mathcal{E}_{\text{train}}$ and tested on $\mathcal{E}_{\text{test}}$ through the binary classification approach resulting in a performance metric such as AP. The train and test data are chronologically split from the same temporal graph which is assumed to be generated through a common causal mechanism, i.e., $\mathcal{E}_{\text{train}} = \mathcal{E}(0, \tau_0)$, and $\mathcal{E}_{\text{test}} = \mathcal{E}(\tau_0, T)$. In light of the experimental setup, we ask the following question:

> Would the model f which is trained on $\mathcal{E}_{\text{train}}$ perform well if tested on a distorted version of $\mathcal{E}_{\text{test}}$ instead of $\mathcal{E}_{\text{test}}$?

To formalise the question in the counterfactual framework (Pearl, 2019), consider the following statements. x' : The model f is tested on $\mathcal{E}_{\text{test}}$, y' : The performance metric is α, x : The model f is tested on a *temporally distorted* version of $\mathcal{E}_{\text{test}}$, and y : The performance metric is less than α. Additionally, y_x is read as y when x. The counterfactual question is framed as $P(y_x \mid x', y')$, i.e.,

> The probability that the performance metric would be less than α had the test data been a temporally distorted version of $\mathcal{E}_{\text{test}}$, given the performance metric was observed to be at least α when the model was tested on $\mathcal{E}_{\text{test}}$.

To answer the question above, we design the *intervention* as graphically depicted in Fig. 1. The TLP model f is trained on the data $\mathcal{E}_{\text{train}}$. The true test data $\mathcal{E}_{\text{test}}$ is temporally distorted through some function $\mathfrak{D}(\cdot)$ resulting in $\mathcal{E}' = \mathfrak{D}(\mathcal{E}_{\text{test}})$. Finally, we test the model f on the true data $\mathcal{E}_{\text{test}}$ and the temporally distorted data $\mathcal{E}'$ and compare the metrics which may result in either of the two scenarios shown in the figure based on which we can comment on the effectiveness of f.

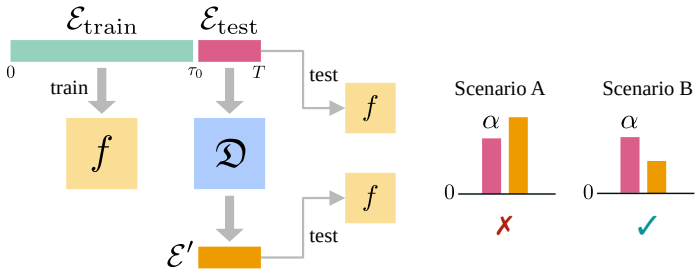

Figure 1: The intervention setup to verify the counterfactual question above.

Consider the following statements,

s_1 : The model f is capable of discerning temporal patterns in $(\mathcal{E}_{\text{train}}, \mathcal{E}_{\text{test}})$
s_2 : The function $\mathfrak{D}$ generates temporally distorted test data $\mathcal{E}' = \mathfrak{D}(\mathcal{E}_{\text{test}})$
s_3 : The data $(\mathcal{E}_{\text{train}}, \mathcal{E}_{\text{test}})$ is learnable
s_4 : The performance metric reported by the model f on true test data $\mathcal{E}_{\text{test}}$ is always higher than that reported on the distorted test data $\mathcal{E}'$, i.e., $P(y_x \mid x', y') = 1$.

We start with $s_1 \wedge s_2 \wedge s_3 \implies s_4$. Assuming that the data is learnable, i.e., $s_3 = 1$, we get $s_1 \wedge s_2 \implies s_4$. Through contraposition, we arrive at $\neg s_4 \implies \neg s_1 \vee \neg s_2$, where $\neg s_4 \equiv P(y_x \mid x', y') \neq 1$. Further, we impose that $\mathfrak{D}$ satisfies $\neg s_2 = 0$, allowing us to conclude $\neg s_4 \implies \neg s_1$ which reads as $P(y_x \mid x', y') \neq 1 \implies$ model f is incapable of discerning the temporal patterns distorted by $\mathfrak{D}$.

Temporal Distortion Techniques We devise distortion functions $\mathfrak{D}(\cdot)$ which enable us to investigate the counterfactual question posed earlier. We propose two distortion techniques $\mathfrak{D}_{\text{INTENSE}}(\cdot, K)$ which creates K time-perturbed copies of each edge event, and $\mathfrak{D}_{\text{SHUFFLE}}(\cdot)$ wherein the timestamps of different edge events are shuffled.

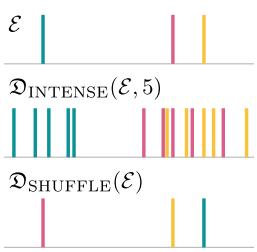

Figure 2: Visual representation of INTENSE and SHUFFLE.

The operations of $\mathfrak{D}_{\text{INTENSE}}$ and $\mathfrak{D}_{\text{SHUFFLE}}$ are described in Algorithms 1 and 2, respectively (see Appendix B). Moreover, a visual example is provided in Fig. 2. In $\mathfrak{D}_{\text{INTENSE}}(\mathcal{E}, 5)$, 5 edge events are created in the vicinity of the true edge event in $\mathcal{E}$. This increases the frequency with which edges appear in an interval, thereby distorting the temporal pattern. In $\mathfrak{D}_{\text{SHUFFLE}}(\mathcal{E})$, as the name suggests, the order in which the edges appear is shuffled and thus the temporal pattern is distorted, the edges now appear where they should not be. The source code of the distortion methods is available here.

3 RESULTS

We evaluate the performance of the following TLP models in light of our counterfactual question: `JODIE` (Kumar et al., 2019), `TGAT` (Xu et al., 2020), `TGN` (Rossi et al., 2020), `CAWN` (Wang et al., 2020), `GraphMixer` (Cong et al., 2023), `DyGFormer` (Yu et al., 2023)

The models are evaluated under two settings: *transductive* and *inductive*. In transductive TLP, the nodes u, v in the positive sample $(u, v, t) \in \mathcal{E}_{\text{test}}$ were observed during training. In contrast, in inductive TLP, at least one node in u, v is novel, and was not observed during training.

In Table 1, we have arranged the datasets in increasing order of their size. We notice that all the TLP models pass the counterfactual test for SHUFFLE distortion on the smallest dataset: `uci`, and some of them {`TGAT`, `GraphMixer`, `DyGFormer`} pass for SHUFFLE on the second-smallest dataset `wikipedia`, and only `GraphMixer` and `TGN` pass on `reddit`. Surprisingly, `JODIE` passes on INTENSE distortion for two of the largest datasets `lastfm` and `mooc`. And overall, none of the TLP models pass the counterfactual test on the INTENSE distortions. This allows us to conclude the following: (1) The TLP models are able to discern the temporal order of edge occurrence, however this capability worsens for larger datasets, and (2) the TLP models do not use the frequencies at which the edges appear over time.

Table 1: Performance of the models JODIE, TGAT, TGN, CAWN, GraphMixer, and DyGFormer on five datasets, and their temporally distorted versions denoted as INTENSE, and SHUFFLE. For each metric, we report the mean, and the 95% confidence interval (CI) as mean ± CI. We have marked the metrics in blue when the counterfactual test fails, and orange when it passes for a given model on a particular dataset.

		AP					AU-ROC				
		uci	wikipedia	reddit	lastfm	mooc	uci	wikipedia	reddit	lastfm	mooc
JODIE	*transductive*	0.8726 ± 5e-3	0.9137 ± 5e-3	0.9654 ± 5e-3	0.7036 ± 2e-3	0.8068 ± 6e-4	0.8950 ± 3e-3	0.9170 ± 3e-3	0.9679 ± 4e-3	0.6798 ± 3e-3	0.8178 ± 4e-3
	INTENSE	0.9129 ± 5e-3	0.9078 ± 1e-2	0.9567 ± 1e-2	0.7090 ± 3e-4	0.7556 ± 4e-4	0.9244 ± 2e-3	0.9177 ± 7e-3	0.9619 ± 9e-3	0.7124 ± 4e-3	0.6014 ± 2e-3
	SHUFFLE	0.8509 ± 3e-3	0.8962 ± 4e-2	0.9613 ± 4e-2	0.7036 ± 1e-3	0.8072 ± 5e-4	0.8852 ± 3e-3	0.9097 ± 2e-2	0.9661 ± 1e-2	0.6798 ± 4e-3	0.8177 ± 5e-3
	inductive	0.7310 ± 2e-2	0.8970 ± 5e-3	0.9138 ± 2e-2	0.8431 ± 4e-3	0.7931 ± 1e-3	0.7546 ± 8e-3	0.8941 ± 4e-3	0.9343 ± 9e-3	0.8091 ± 2e-3	0.8363 ± 1e-3
	INTENSE	0.8332 ± 8e-3	0.8972 ± 1e-2	0.9308 ± 4e-2	0.8361 ± 5e-4	0.7658 ± 3e-4	0.8384 ± 3e-3	0.9036 ± 1e-2	0.9457 ± 3e-2	0.8302 ± 1e-3	0.6912 ± 6e-3
	SHUFFLE	0.6994 ± 8e-3	0.9078 ± 2e-2	0.9251 ± 6e-3	0.8431 ± 2e-3	0.7931 ± 7e-4	0.7368 ± 5e-3	0.9157 ± 1e-2	0.9419 ± 3e-2	0.8091 ± 2e-3	0.8363 ± 1e-3
TGAT	*transductive*	0.7694 ± 7e-3	0.9528 ± 2e-3	0.9818 ± 6e-4	0.7309 ± 3e-4	0.8458 ± 5e-4	0.7885 ± 1e-2	0.9499 ± 2e-3	0.9806 ± 6e-4	0.7139 ± 4e-4	0.8587 ± 2e-4
	INTENSE	0.8637 ± 2e-2	0.9691 ± 3e-3	0.9825 ± 6e-4	0.9840 ± 1e-4	0.9610 ± 1e-4	0.8707 ± 1e-2	0.9680 ± 2e-3	0.9821 ± 6e-4	0.9835 ± 1e-4	0.9627 ± 1e-4
	SHUFFLE	0.7336 ± 2e-2	0.9532 ± 5e-3	0.9826 ± 6e-3	0.7308 ± 3e-4	0.8458 ± 4e-4	0.7719 ± 1e-2	0.9492 ± 5e-3	0.9814 ± 7e-3	0.7139 ± 2e-4	0.8588 ± 4e-4
	inductive	0.7008 ± 1e-2	0.9401 ± 2e-3	0.9658 ± 1e-3	0.7817 ± 2e-4	0.8430 ± 3e-4	0.7020 ± 8e-3	0.9353 ± 2e-3	0.9641 ± 1e-3	0.7661 ± 1e-4	0.8563 ± 2e-4
	INTENSE	0.8095 ± 2e-2	0.9621 ± 2e-3	0.9676 ± 1e-3	0.9841 ± 1e-4	0.9621 ± 1e-4	0.8019 ± 2e-2	0.9604 ± 2e-3	0.9676 ± 8e-4	0.9837 ± 2e-4	0.9628 ± 1e-4
	SHUFFLE	0.6324 ± 1e-2	0.9304 ± 7e-3	0.9664 ± 3e-3	0.7817 ± 2e-4	0.8430 ± 3e-4	0.6558 ± 7e-3	0.9257 ± 7e-3	0.9644 ± 7e-3	0.7661 ± 2e-4	0.8563 ± 2e-4
TGN	*transductive*	0.7975 ± 1e-2	0.9472 ± 1e-3	0.9578 ± 1e-3	0.7764 ± 5e-3	0.8855 ± 4e-3	0.7826 ± 1e-2	0.9370 ± 1e-3	0.9545 ± 1e-3	0.6246 ± 6e-3	0.8196 ± 7e-4
	INTENSE	0.9709 ± 3e-3	0.9911 ± 6e-4	0.9744 ± 2e-3	0.9916 ± 1e-5	0.9629 ± 6e-4	0.9653 ± 3e-3	0.9898 ± 1e-3	0.9723 ± 2e-3	0.9266 ± 5e-4	0.9222 ± 2e-4
	SHUFFLE	0.6520 ± 4e-2	0.8487 ± 3e-2	0.9563 ± 2e-3	0.7764 ± 9e-4	0.8848 ± 1e-3	0.6722 ± 6e-2	0.8310 ± 3e-2	0.9533 ± 2e-3	0.6246 ± 5e-3	0.8197 ± 2e-3
	inductive	0.7948 ± 6e-3	0.9463 ± 1e-3	0.9346 ± 1e-3	0.8336 ± 4e-3	0.8873 ± 1e-3	0.7714 ± 6e-3	0.9374 ± 1e-3	0.9299 ± 1e-3	0.6935 ± 4e-3	0.8033 ± 3e-3
	INTENSE	0.9650 ± 2e-3	0.9908 ± 6e-4	0.9645 ± 3e-3	0.9927 ± 2e-5	0.9641 ± 2e-4	0.9592 ± 3e-3	0.9903 ± 1e-3	0.9617 ± 3e-3	0.9374 ± 4e-4	0.9144 ± 2e-4
	SHUFFLE	0.6193 ± 9e-3	0.8376 ± 3e-2	0.9299 ± 3e-3	0.8337 ± 6e-3	0.8872 ± 2e-3	0.6245 ± 2e-2	0.8194 ± 2e-2	0.9266 ± 4e-3	0.6936 ± 4e-3	0.8033 ± 3e-3
CAWN	*transductive*	0.9397 ± 8e-4	0.9901 ± 1e-4	0.9884 ± 3e-3	0.8755 ± 3e-4	0.8667 ± 2e-4	0.9162 ± 9e-4	0.9886 ± 1e-4	0.9864 ± 4e-3	0.8494 ± 3e-4	0.8653 ± 2e-4
	INTENSE	0.9889 ± 7e-4	0.9975 ± 8e-5	0.9942 ± 7e-5	0.9879 ± 2e-4	0.9719 ± 1e-4	0.9848 ± 6e-4	0.9977 ± 9e-5	0.9931 ± 8e-5	0.9871 ± 1e-4	0.9734 ± 1e-4
	SHUFFLE	0.8866 ± 2e-3	0.9887 ± 3e-4	0.9880 ± 2e-3	0.8755 ± 3e-4	0.8666 ± 3e-4	0.8495 ± 7e-3	0.9868 ± 3e-4	0.9859 ± 6e-4	0.8494 ± 3e-4	0.8653 ± 4e-4
	inductive	0.9273 ± 2e-3	0.9896 ± 4e-4	0.9859 ± 3e-3	0.9031 ± 5e-4	0.8543 ± 4e-4	0.9052 ± 1e-2	0.9877 ± 5e-4	0.9833 ± 5e-3	0.8822 ± 4e-4	0.8519 ± 3e-4
	INTENSE	0.9857 ± 2e-3	0.9971 ± 1e-5	0.9938 ± 8e-5	0.9889 ± 3e-4	0.9731 ± 2e-4	0.9810 ± 3e-3	0.9972 ± 6e-4	0.9929 ± 8e-5	0.9882 ± 1e-4	0.9737 ± 1e-4
	SHUFFLE	0.8783 ± 3e-2	0.9896 ± 6e-3	0.9851 ± 1e-3	0.9030 ± 5e-4	0.8541 ± 4e-4	0.8383 ± 3e-2	0.9876 ± 1e-2	0.9826 ± 8e-4	0.8822 ± 4e-4	0.8518 ± 2e-4
GraphMixer	*transductive*	0.9323 ± 2e-3	0.9690 ± 4e-4	0.9738 ± 3e-4	0.7630 ± 1e-4	0.8233 ± 3e-4	0.9176 ± 2e-3	0.9654 ± 7e-4	0.9727 ± 3e-4	0.7406 ± 1e-4	0.8363 ± 2e-4
	INTENSE	0.9923 ± 6e-4	0.9966 ± 2e-4	0.9965 ± 1e-4	0.9858 ± 1e-4	0.9537 ± 1e-4	0.9916 ± 5e-4	0.9968 ± 1e-4	0.9969 ± 1e-4	0.9856 ± 1e-4	0.9590 ± 1e-4
	SHUFFLE	0.8553 ± 3e-3	0.9096 ± 1e-3	0.9725 ± 2e-4	0.7630 ± 1e-4	0.8230 ± 2e-4	0.8476 ± 3e-3	0.9062 ± 3e-4	0.9712 ± 3e-4	0.7406 ± 1e-4	0.8361 ± 2e-4
	inductive	0.9133 ± 1e-3	0.9639 ± 1e-4	0.9517 ± 8e-4	0.8261 ± 3e-4	0.8077 ± 2e-4	0.8960 ± 2e-3	0.9600 ± 2e-4	0.9489 ± 9e-4	0.8065 ± 2e-4	0.8224 ± 2e-4
	INTENSE	0.9771 ± 5e-4	0.9939 ± 1e-4	0.9937 ± 2e-4	0.9867 ± 1e-4	0.9555 ± 1e-4	0.9779 ± 1e-4	0.9946 ± 1e-4	0.9947 ± 2e-4	0.9864 ± 1e-4	0.9592 ± 1e-4
	SHUFFLE	0.7945 ± 3e-4	0.8900 ± 2e-3	0.9477 ± 7e-4	0.8261 ± 3e-4	0.8072 ± 3e-4	0.7869 ± 3e-4	0.8815 ± 2e-3	0.9447 ± 1e-3	0.8065 ± 2e-4	0.8222 ± 3e-4
DyGFormer	*transductive*	0.9596 ± 3e-4	0.9901 ± 2e-4	0.9921 ± 1e-4	0.9096 ± 1e-4	0.8622 ± 2e-4	0.9478 ± 5e-4	0.9890 ± 3e-4	0.9913 ± 1e-4	0.8959 ± 3e-4	0.8622 ± 1e-4
	INTENSE	0.9938 ± 1e-4	0.9983 ± 1e-4	0.9984 ± 1e-4	0.9912 ± 1e-4	0.9709 ± 1e-4	0.9924 ± 1e-4	0.9986 ± 1e-4	0.9988 ± 1e-4	0.9911 ± 1e-4	0.9728 ± 1e-4
	SHUFFLE	0.9515 ± 1e-3	0.9892 ± 1e-4	0.9924 ± 1e-4	0.9096 ± 2e-4	0.8620 ± 4e-4	0.9391 ± 8e-4	0.9875 ± 1e-4	0.9915 ± 1e-4	0.8959 ± 3e-4	0.8622 ± 3e-4
	inductive	0.9437 ± 1e-4	0.9854 ± 5e-4	0.9880 ± 3e-4	0.9293 ± 1e-4	0.8509 ± 3e-4	0.9241 ± 1e-4	0.9845 ± 4e-4	0.9866 ± 3e-4	0.9180 ± 2e-4	0.8529 ± 2e-4
	INTENSE	0.9854 ± 1e-4	0.9965 ± 4e-5	0.9973 ± 1e-5	0.9918 ± 2e-4	0.9723 ± 1e-4	0.9831 ± 1e-4	0.9976 ± 2e-4	0.9981 ± 1e-4	0.9916 ± 1e-4	0.9734 ± 1e-4
	SHUFFLE	0.9291 ± 4e-4	0.9833 ± 3e-4	0.9878 ± 3e-4	0.9293 ± 1e-4	0.8506 ± 5e-4	0.9057 ± 6e-4	0.9812 ± 2e-4	0.9866 ± 3e-4	0.9180 ± 2e-4	0.8528 ± 3e-4

Discussion Some of the TLP models used in this work such as GraphMixer, and DyGFormer are considered the SoTA on most datasets, with near-perfect performance. However, as we showed earlier, a higher metric alone is not indicative of good performance without sanity checks. The counterfactual question helps make the evaluation more explainable, as models that perform worse on temporally distorted data can claim superiority over models that do not. An ideal TLP model should be able to capture the difference in the count of edge events, their order, and the temporal shifts in the edge events.

To reiterate, if the performance of the model on the temporally distorted test data is similar or better than the performance on the original test data, then it implies one the following: (a) the model has not made use of the temporal information in the training set, (b) there is no useful temporal information in the dataset, or (c) the temporal distortion is weak. In the absence of a guarantee that the dataset has useful temporal information that can aid prediction, we can compare different models through the performance gaps.

Future Work Moving away from the binary classification approach to assess the performance of temporal link prediction, future research should explore a generative approach where after observing a temporal graph from time $t \in (0, \tau_0)$, the model can generate a temporal graph in $t \in (\tau_0, T)$. This generated temporal graph can then be compared with the ground truth to measure similarity and assess the performance of the model.

Conclusion In this work, instead of introducing novel datasets, we present techniques for generating temporally distorted versions of any temporal graph dataset. This makes the contribution relevant even for datasets which will be introduced in the future. To the best of our knowledge, we are the first to apply counterfactual analysis to TLP and hope that it can help standardize the assessment of TLP models.

REFERENCES

W. Cong, S. Zhang, J. Kang, B. Yuan, H. Wu, X. Zhou, H. Tong, and M. Mahdavi. Do We Really Need Complicated Model Architectures For Temporal Networks? In *The Eleventh International Conference on Learning Representations*, Sept. 2023.

N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, and N. B. Anuar. Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications*, 166:102716, 2020.

W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

Z. Fan, Z. Liu, J. Zhang, Y. Xiong, L. Zheng, and P. S. Yu. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 433–442, 2021.

S. Huang, F. Poursafaei, J. Danovitch, M. Fey, W. Hu, E. Rossi, J. Leskovec, M. Bronstein, G. Rabusseau, and R. Rabbany. Temporal graph benchmark for machine learning on temporal graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

Y. Kim, Y. Lee, M. Choe, S. Oh, and Y. Lee. Temporal graph networks for graph anomaly detection in financial networks. *arXiv preprint arXiv:2404.00060*, 2024.

S. Kumar, X. Zhang, and J. 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, 2019.

P. Panzarasa, T. Opsahl, and K. M. Carley. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5):911–932, 2009.

J. Pearl. *Causality*. Cambridge university press, 2009.

J. Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019.

J. W. Pennebaker, M. E. Francis, and R. J. Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

F. Poursafaei and R. Rabbany. Exhaustive Evaluation of Dynamic Link Prediction. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1121–1130, Shanghai, China, Dec. 2023. IEEE. ISBN 9798350381641. doi: 10.1109/ICDMW60847.2023.00147.

F. Poursafaei, S. Huang, K. Pelrine, and R. Rabbany. Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems*, 35:32928–32941, 2022.

M. Qin and D.-Y. Yeung. Temporal Link Prediction: A Unified Framework, Taxonomy, and Review, June 2023.

Y. Qin, W. Ju, H. Wu, X. Luo, and M. Zhang. Learning graph ode for continuous-time sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.

Y. Wang, Y.-Y. Chang, Y. Liu, J. Leskovec, and P. Li. Inductive representation learning in temporal networks via causal anonymous walks. In *International Conference on Learning Representations*, 2020.

D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations*, 2020.

L. Yu, L. Sun, B. Du, and W. Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023.

A DATASETS & MODELS

A.1 TEMPORAL GRAPH DATASETS

We use the following datasets[1] to perform counterfactual analysis[2]:

- `wikipedia` (Kumar et al., 2019) describes a dynamic graph of interaction between the editors and Wikipedia pages over a span of one month. The entries consist of the user ID, page ID, and timestamp. The edge features are LIWC-feature vectors (Pennebaker et al., 2001) of the edit text. The edge feature dimension is 172.
- `reddit` (Kumar et al., 2019) describes a bipartite interaction graph between the users and subreddits. The interaction event is recorded with the IDs of the user, subreddit and timestamp. Similar to `wikipedia`, the post content is converted into a LIWC-feature vector of dimension 172 which serves as the edge feature.
- `uci` (Panzarasa et al., 2009) is a dynamic graph describing message-exchange among the students at University of California at Irvine (UCI) from April to October 2004. The interaction event consists of the user IDs, and timestamp.
- `lastfm` (Kumar et al., 2019) is also a bipartite graph depicting the interactions between 1000 users and 1000 most listened songs over a span of one month.
- `mooc` (Kumar et al., 2019) as the name suggests is a student interaction network enrolled in the same online course.

Table 2: The scale of different datasets.

Dataset	Total nodes (10^3)	Total Edges (10^3)	Unique Edges (10^3)
`uci`	1.89	59.84	20.29
`wikipedia`	9.23	157.47	18.25
`reddit`	10.98	672.45	78.52
`lastfm`	1.98	1293.10	154.99
`mooc`	7.14	411.75	178.44

A.2 TEMPORAL LINK PREDICTION MODELS

We make use of the following models[3] to test the counterfactual framework:

- `JODIE` (Kumar et al., 2019) uses a recurrent neural network (RNN) to generate node embeddings for each interaction event. The future embedding of a node is estimated through a novel projection operator which is turn in used to predict future edge events.
- `TGAT` (Xu et al., 2020) relies on self-attention mechanism to generate node embeddings to capture the temporal evolution of the graph structure.
- `TGN` (Rossi et al., 2020) combine memory modules with graph-based operators to create an encoder-decoder pair capable of creating temporal node embeddings.
- `CAWN` (Wang et al., 2020) propose a novel strategy based on the law of triadic closure, where temporal walks retrieve the dynamic graph motifs without explicitly counting and selecting the motifs. The node IDs are replaced with the hitting counts to facilitate inductive inference.
- `GraphMixer` (Cong et al., 2023) use a simple architecture where the encoder and decoder are designed using multi-layer perceptrons (MLPs).
- `DyGFormer` (Yu et al., 2023) use a transformer to learn from nodes' first-hop interactions and report SoTA results on most of the datasets.

[1]The datasets can be downloaded from https://zenodo.org/records/7213796

[2]The datasets are chronologically split in the ratio $0.7 : 0.15 : 0.15$ into train, validation, and test sets.

[3]The optimal hyper-parameters reported by the models are used.

B TEMPORAL DISTORTION ALGORITHMS

Algorithm 1 $\mathfrak{D}_{\text{INTENSE}}$

Input $\mathcal{E}, K \in \mathbb{N}, \bar{\tau} \in \mathbb{R}^+$
Output $\mathcal{E}'$

$\mathcal{E}' = \varnothing$
for $(u, v, t) \in \mathcal{E}$ **do**
  for $k \in [K]$ **do**
    $\tau \sim \text{Uniform}(-\bar{\tau}, \bar{\tau})$
    $\mathcal{E}' \leftarrow \mathcal{E}' \cup \{(u, v, t + \tau)\}$
  end for
end for

Algorithm 2 $\mathfrak{D}_{\text{SHUFFLE}}$

Input $\mathcal{E}$
Output $\mathcal{E}'$

$\mathcal{E}' = \varnothing$
$\mathcal{T} \leftarrow \mathscr{T}(\mathcal{E})$
for $(u, v, t) \in \mathcal{E}$ **do**
  $\tau \sim \mathcal{T}$
  $\mathcal{E}' \leftarrow \mathcal{E}' \cup \{(u, v, \tau)\}$
  $\mathcal{T} \leftarrow \mathcal{T} \setminus \{\tau\}$
end for