BenchTemp: A General Benchmark for Evaluating Temporal Graph Neural Networks

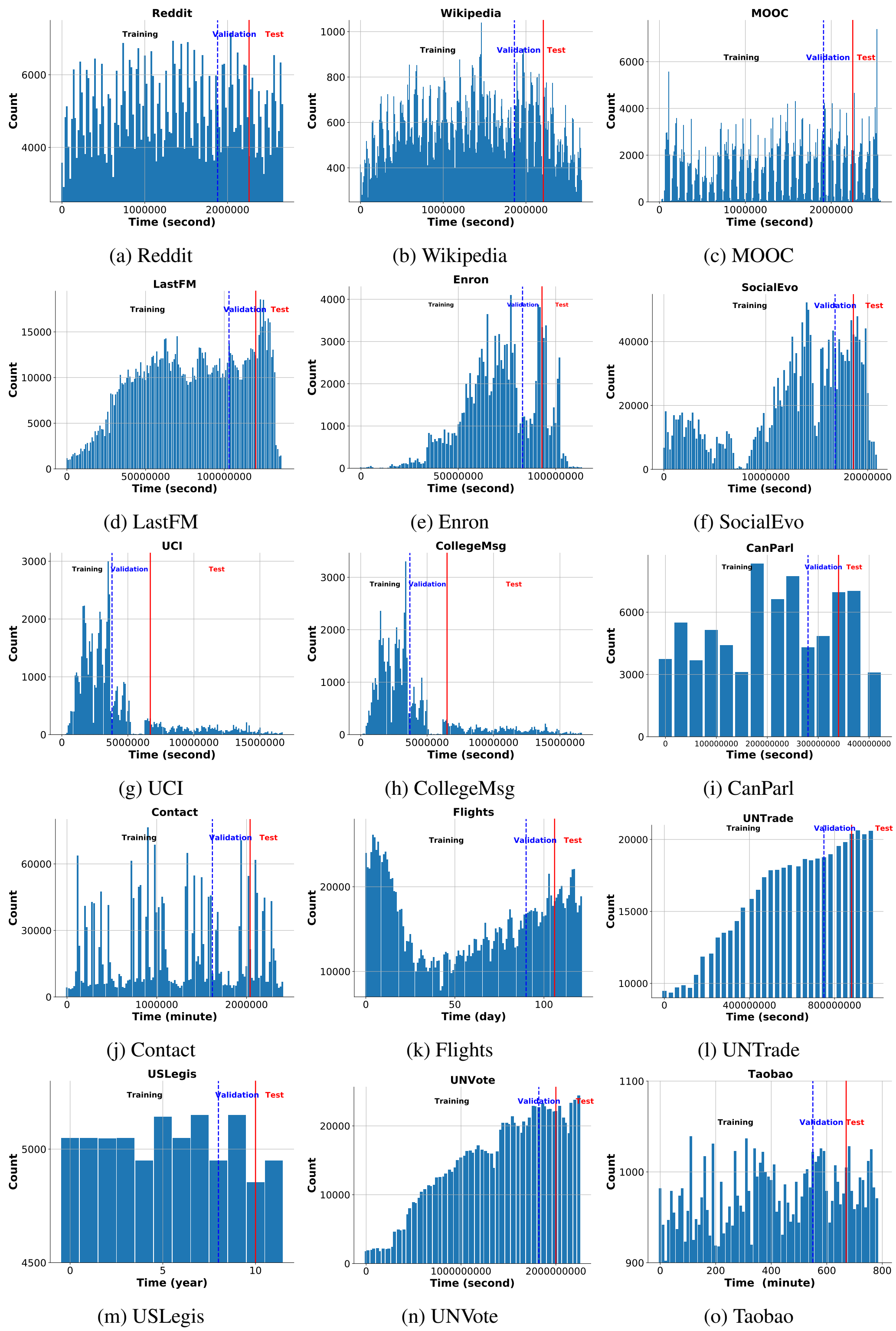
Appendix

A Temporal Graph Datasets

As shown in Table 2 of the main paper, we select fifteen benchmark databases from diverse domains. All datasets are publicly available under CC BY-NC licence and can be accessed at https://drive.google.com/drive/folders/1HKSFGEfxHDlHuQZ6nK4SLCEMFQIOtzpz?usp=sharing. Figure 1 shows the temporal distribution of edges in the evaluated temporal graphs.

- Reddit is a bipartite interaction graph, consisting of one month of posts made by users on subreddits [1]. Users and subreddits are nodes, and egdes are interactions of users writing posts to subreddits. The text of each post is converted into LIWC-feature vector [2] as an edge feature of length 172. This public dataset gives 366 true labels among 672,447 interactions, and those true label are ground-truth labels of banned users from Reddit [3].
- Wikipedia is a bipartite interaction graph, and contains one month of edits made by editors. This public dataset selects the 1,000 most edited pages as items and editors who made at least 5 edits as users over a month [3]. Editors and pages are nodes, and edges are interactions of editors editing on pages. Edge features of length 172 are interaction edits converted into LIWC-feature vectors [2]. Wikipedia dataset treats 217 public ground-truth labels of banned users from 157,474 interactions as positive labels.
- MOOC is a bipartite MOOC online network of students and online course content units [4]. Students and courses are nodes, and edges with features of length 4 are interactions of user viewing a video, submitting an answer, etc. This public dataset treats 4,066 dropout events out of 411,749 interactions as positive labels [3].
- LastFM is a user-song bipartite network [5]. Users and songs are nodes, and edges are user-listens-song interactions. This public dataset includes 1,293,103 interactions between all 1000 users and the 1000 most listened songs [3].
- Enron is an email communication network that collects about half a million emails over several years [6]. Nodes of the network are email addresses, and edges are email communication between accounts [7].
- SocialEvo is a network in which experiments are conducted to closely track the everyday life of a whole undergraduate dormitory with mobile phones. This public dataset is collected by a cell phone application every six minutes, and contains physical proximity and location between students living in halls of residence. [8].
- UCI is a facebook-like social network that contains user posts to forums. Nodes are students (1,899) at University of California, Irvine, and edges are interactions of online messages (59,835) among these users [9]. Each edge has 100 features.
- CollegeMsg is provided by the SNAP team of Stanford [10]. This dataset is derived from the facebook-like social network introduced in dataset UCI. The SNAP team has parsed it to a temporal network. Each edge has 172 features.

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.



(a) Reddit (b) Wikipedia (c) MOOC

(d) LastFM (e) Enron (f) SocialEvo

(g) UCI (h) CollegeMsg (i) CanParl

(j) Contact (k) Flights (l) UNTrade

(m) USLegis (n) UNVote (o) Taobao

Figure 1: The temporal distribution of edges of the evaluated temporal graphs.

- CanParl is a Canadian parliament bill voting network extracted from open website [11]. Nodes are members of parliament (MPs), and edges are the interactions between MPs from 2006 to 2019.

- Contact is a temporal and weighted network of physical proximity among the participants [12]. Nodes are participant and edges are proximity events between the study participants. Edge features indicate the physical proximity between participants [13].
- Flights is a weighted flight network. Nodes are airports, and edges are tracked flights [14]. The weights of edges indicate the number of flights between two given airports within a day [13].
- UNTrade is a food and agriculture trading weighted network among 181 nations over 30 years [15]. Nodes are countries, and edges are tradings between two countries. The weights of edges are the total sum of normalized agriculture import or export values between two given countries [13].
- USLegis is a senate co-sponsorship network that examines the social relations of legislators in their co-sponsorship relationships on bills [11]. Nodes are congress members, and edge weights are the number of times that two members of congress co-sponsor a bill in a given congress [13].
- UNVote is a weighted network of roll-call votes in the UN General Assembly 1946-2021 [16]. Nodes are nations, and edge weights are the number of times both nations have voted "yes" to an item.
- Taobao is a subset of the Taobao user behavior dataset intercepted based on the period 8:00 to 18:00 on 26 November 2017 [17]. This public dataset is a user-item bipartite network. Nodes are users and items, and edges are behaviors between users and items, such as favor, click, purchase, and add an item to shopping cart. Each edge has 4 features, corresponding to 4 different types of behaviors [18].

B Experiment Details

DataLoader of the link prediction pipeline introduced in Section 3.2.1 splits and generates training set, validation set, transductive set, and inductive test sets depending on the New-Old and New-New settings, for link prediction task. The detailed statistics of these data sets are shown in Table 1. DataLoader of the node classification pipeline introduced in Section 3.2.2 follows the traditional transductive setting. The detailed statistics of the training set, validation set, and test set on three available datasets (Reddit, Wikipedia, and MOOC) are given in Table 2.

EdgeSampler of link prediction pipeline introduced in Section 3.2.1 uses fixed seeds for different validation sets and test sets to ensure that the test results are reproducible across different runs.

C Model Implementation Details

We implement JODIE, DyRep, and TGN based on the TGN framework. Furthermore, we fix the inconsistencies of implementations between link prediction task and node classification task.

TGAT concatenates *node features*, *edge features*, *time features*, and *position features* to perform the multi-head self-attention mechanism. There is a positional encoding in the self-attention mechanism for capturing sequential information. Let d_n, d_e, d_{time}, and d_{pos} denote the dimensions of node features, edge features, time features, and positional encoding, respectively. The number of attention heads is n_{head}. These parameters must satisfy:

$$(d_n + d_e + d_{time} + d_{pos})\%n_{head} = 0 \tag{1}$$

The experimental parameters of TGAT are summarized in Table 3.

Similar to the setup of TGAT, CAWN adopts a multi-head self-attention mechanism to capture the subtle relevance of *node features*, *edge features*, *time features*, and *positional features*. Those parameters satisfy Formula (1) as well, and $d_n = d_{time}$. However, CAWN initializes the number of attention heads to 2, so we change the dimension of d_{pos} to conduct experiments. The experimental parameters of CAWN are shown in Table 4.

Table 1: Statistics of datasets for link prediction task. "New-Old Validation" indicates the validation set under Inductive New-Old setting, and so on.

	Training		**Validation**		**Transductive Test**		**Inductive Validation**		**Inductive Test**	
	# nodes	*# edges*	*# nodes*	*# edges*	*# nodes*	*# edges*	*# nodes*	*# edges*	*# nodes*	*# edges*
Reddit	9,574	389,989	9,839	100,867	9,615	100,867	3,491	19,446	3,515	21,470
Wikipedia	6,141	81,029	3,256	23,621	3,564	23,621	2,120	12,016	2,437	11,715
MOOC	6,015	227,485	2,599	61,762	2,412	61,763	2,333	25,592	2,181	29,179
LastFM	1,612	722,758	1,714	193,965	1,753	193,966	1,643	57,651	1,674	98,442
Enron	157	79,064	155	18,786	141	18,785	112	5,637	110	4,859
SocialEvo	67	1,222,980	64	314,930	62	314,924	62	62,811	60	70,038
UCI	1,338	34,386	1,036	8,975	847	8,976	816	4,761	678	5,707
CollegeMsg	1,337	34,544	1,036	8,975	847	8,975	818	4,914	680	5,885
CanParl	618	47,435	344	11,809	342	10,113	344	5,481	341	5,591
Contact	617	1,372,030	632	364,005	629	363,780	582	68,261	590	69,617
Flights	11,230	1,107,798	10,844	279,399	10,906	287,824	6,784	54,861	6,820	58,102
UNTrade	230	291,287	230	78,721	228	61,595	227	17,528	226	14,001
USLegis	176	38,579	113	10,005	100	4,950	113	5,010	100	3,297
UNVote	178	600,511	194	135,298	194	155,119	194	28,136	194	33,083
Taobao	54,462	45,630	17,964	11,621	18,143	11,550	16,476	10,338	16,896	10,516

	New-Old Validation		**New-Old Test**		**New-New Validation**		**New-New Test**		**Unseen Nodes**
	# nodes	*# edges*	*# nodes*	*# edges*	*# nodes*	*# edges*	*# nodes*	*# edges*	
Reddit	3,301	16,760	3,325	18,703	488	2,686	486	2,767	1,098
Wikipedia	1,809	8,884	1,996	8,148	468	3,132	629	3,567	922
MOOC	2,316	23,109	2,164	25,730	553	2,483	592	3,449	714
LastFM	1,642	52,379	1,674	63,505	272	5,272	331	34,937	198
Enron	111	4,965	109	4,262	19	672	20	597	18
SocialEvo	62	58,959	60	65,466	7	3,852	7	4,572	7
UCI	757	3,686	606	4,193	247	1,075	213	1,514	189
CollegeMsg	759	3,839	608	4,328	247	1,075	214	1,557	189
CanParl	344	4,543	341	4,469	106	938	111	1,122	73
Contact	582	64,887	590	65,883	62	3,374	59	3,734	69
Flights	6,711	49,796	6,739	52,504	874	5,065	937	5,598	1,316
UNTrade	227	16,420	226	13,112	25	1,108	25	889	25
USLegis	112	4,154	100	2,436	37	856	42	861	22
UNVote	194	26,545	194	31,166	23	1,591	23	1,917	20
Taobao	9,247	5,678	8,136	4,860	7,706	4,660	9,298	5,656	8,256

Table 2: Statistics of datasets for node classification task.

	Training		**Validation**		**Test**	
	# nodes	*# edges*	*# nodes*	*# edges*	*# nodes*	*# edges*
Reddit	10,844	470,713	9,839	100,867	9,615	100,867
Wikipedia	7,475	110,232	3,256	23,621	3,564	23,621
MOOC	6,625	288,224	2,599	61,762	2,412	61,763

Table 3: Experimental parameters of TGAT.

	d_n	d_e	d_{time}	d_{pos}	n_{head}
Reddit	172	172	172	172	2
Wikipedia	172	172	172	172	2
MOOC	172	4	172	172	2
LastFM	172	2	172	172	2
Enron	172	32	172	172	2
SocialEvo	172	2	172	172	2
UCI	172	100	172	172	2
CollegeMsg	172	172	172	172	2
CanParl	172	1	172	172	1
Contact	172	1	172	172	1
Flights	172	1	172	172	1
UNTrade	172	1	172	172	1
USLegis	172	1	172	172	1
UNVote	172	1	172	172	1
Taobao	172	4	172	172	2

NeurTW concatenates *node features*, *edge features*, and *positional features* (without *time features*) during the temporal random walk encoding. Regarding the temporal walk sampling strategy, given a

Table 4: Experimental parameters of CAWN.

	d_n	d_e	d_{time}	d_{pos}
Reddit	172	172	172	108
Wikipedia	172	172	172	108
MOOC	172	4	172	100
LastFM	172	2	172	102
Enron	172	32	172	104
SocialEvo	172	2	172	102
UCI	172	100	172	100
CollegeMsg	172	172	172	108
CanParl	172	1	172	103
Contact	172	1	172	103
Flights	172	1	172	103
UNTrade	172	1	172	103
USLegis	172	1	172	103
UNVote	172	1	172	103
Taobao	172	4	172	100

node u at time t, the sampling probability weight of its neighbor v $((\{v,u\},t') \in \mathcal{G}_{u,t})$ is proportion to $exp(\alpha(t'-t))$, where α is a temporal bias. This sampling strategy is a temporal-biased sampling method. However, the time intervals in some benchmark datasets (Enron, CanParl, UNTrade, USLegis, and UNVote) are relatively large, and the exponential sampling probability weights may encounter overflow. Therefore, we propose a strategy to calculate the sampling probability weights for these datasets:

$$W(v,t') = \begin{cases} t'-t, & t'-t>0, \\ 1, & t'-t=0, \\ -1/(t'-t), & t'-t<0, \end{cases} \tag{2}$$

where $W(v,t')>0$. This strategy can avoid overflow and is also a temporal-biased sampling method. Finally, the sampling probability of each neighbor is obtained after normalization:

$$Pr_t(v) = \frac{\alpha W(v,t_v)}{\sum_{v' \in \mathcal{G}_{u,t}} \alpha W(v',t_{v'})}, \tag{3}$$

where α is a temporal bias. For other hyperparameters that we have not mentioned, we use default values from the original experiments in the corresponding papers. All the experimental codes are publicly available under MIT license and can be accessed at `https://github.com/qianghuangwhu/benchtemp`.

D Experiment Results

D.1 AP Results for Link Prediction

We show the average precision (AP) results on link prediction task and highlight the best and second-best numbers for each job in Table 5. The overall performance is similar to that of AUC. For the transductive setting, CAWN gives impressive results and achieves the best or second-best results on 12 datasets out of 15, followed by NeurTW (7 out of 15), TGN (5 out of 15), and NAT (5 out of 15), verifying the effectiveness of temporal walk, temporal memory, and joint neighborhood on transductive link prediction task. For the inductive setting, CAWN and NAT both rank top-2 on 9 datasets, followed by NeurTW on 6. Results reveal that models based on temporal walks and joint neighborhood can better capture structure patterns on edges that have never been seen. TGN performs relatively poorly for the inductive link prediction task on almost all datasets. We can draw similar conclusions from inductive New-Old and inductive New-Old experimental results. We note that DyRep achieves the best AP result under the inductive New-Old setting and the second-best AP result under the inductive setting on the SocialEvo dataset. SocialEvo has the maximum average degree and edge density as shown in Table 2 in the main paper, demonstrating that DyRep performs better for inductive link prediction tasks on dense temporal graphs.

Table 5: Average Precision (AP) results on link prediction task. "*" denotes that the model encounters runtime error; "—" denotes timeout after 48 hours. The best and second-best results are highlighted as **bold red** and underlined blue. Some standard deviations are zero because we terminate those models that can only run one epoch within 2 days. We do not highlight the second-best if the gap is > 0.05 compared with the best result.

Transductive							
Dataset \ Model	JODIE	DyRep	TGN	TGAT	CAWN	NeurTW	NAT
Reddit	0.9718 ± 0.0022	0.9808 ± 0.0006	0.9874 ± 0.0002	0.9822 ± 0.0003	**0.9904 ± 0.0001**	0.9855 ± 0.0013	0.9868 ± 0.0017
Wikipedia	0.9471 ± 0.0056	0.9464 ± 0.0010	0.9852 ± 0.0003	0.9536 ± 0.0022	0.9906 ± 0.0001	**0.9918 ± 0.0001**	0.9819 ± 0.0026
MOOC	0.7364 ± 0.0370	0.7933 ± 0.0348	0.883 ± 0.0242	0.7185 ± 0.0051	**0.9369 ± 0.0009**	0.7943 ± 0.0248	0.7537 ± 0.0191
LastFM	0.6762 ± 0.0678	0.6736 ± 0.0768	0.7694 ± 0.0276	0.5375 ± 0.0044	**0.8946 ± 0.0006**	0.8405 ± 0.0	0.8729 ± 0.0022
Enron	0.7841 ± 0.0254	0.7648 ± 0.0418	0.8472 ± 0.0173	0.6063 ± 0.0194	**0.9142 ± 0.0052**	0.8847 ± 0.0079	0.9044 ± 0.0036
SocialEvo	0.7982 ± 0.0476	0.8816 ± 0.0042	**0.9325 ± 0.0006**	0.7724 ± 0.0052	0.9188 ± 0.0011	—	0.8989 ± 0.0096
UCI	0.8436 ± 0.0110	0.4913 ± 0.0367	0.8914 ± 0.0138	0.779 ± 0.0052	0.9425 ± 0.001	**0.9702 ± 0.0021**	0.9253 ± 0.0083
CollegeMsg	0.5276 ± 0.0493	0.5070 ± 0.0049	0.8418 ± 0.0847	0.7902 ± 0.0033	0.9401 ± 0.0025	**0.9727 ± 0.0001**	0.9241 ± 0.0086
CanParl	0.7030 ± 0.0077	0.6860 ± 0.0256	0.6765 ± 0.0615	0.6811 ± 0.0157	0.6916 ± 0.0546	**0.8528 ± 0.0213**	0.6593 ± 0.0764
Contact	0.9087 ± 0.0114	0.9016 ± 0.0319	0.9699 ± 0.0045	0.5888 ± 0.0065	0.9677 ± 0.0024	**0.9756 ± 0.0**	0.945 ± 0.0168
Flights	0.9389 ± 0.0075	0.8836 ± 0.0078	0.9764 ± 0.0025	0.899 ± 0.0025	**0.9860 ± 0.0002**	0.9321 ± 0.0	0.9749 ± 0.0048
UNTrade	0.6329 ± 0.0102	0.6099 ± 0.0057	0.6059 ± 0.0086	*	0.7488 ± 0.0005	0.5648 ± 0.0167	**0.7514 ± 0.0615**
USLegis	0.7585 ± 0.0032	0.6808 ± 0.0368	0.7398 ± 0.0027	0.7206 ± 0.0071	0.9682 ± 0.0048	**0.9713 ± 0.0013**	0.7425 ± 0.016
UNVote	0.6090 ± 0.0076	0.5855 ± 0.0225	**0.6694 ± 0.0095**	0.5388 ± 0.002	0.6175 ± 0.0013	0.6008 ± 0.0	0.6449 ± 0.033
Taobao	0.808 ± 0.0015	0.8074 ± 0.0014	0.8618 ± 0.0004	0.5508 ± 0.0093	0.7464 ± 0.0027	0.8808 ± 0.0012	**0.8933 ± 0.0007**

Inductive							
Dataset \ Model	JODIE	DyRep	TGN	TGAT	CAWN	NeurTW	NAT
Reddit	0.9427 ± 0.0118	0.9582 ± 0.0003	0.9767 ± 0.0003	0.9667 ± 0.0003	0.9889 ± 0.0001	0.9821 ± 0.0006	**0.9912 ± 0.0027**
Wikipedia	0.9316 ± 0.0049	0.9181 ± 0.0037	0.9791 ± 0.0004	0.9389 ± 0.0035	0.9903 ± 0.0002	0.9912 ± 0.0004	**0.9962 ± 0.0021**
MOOC	0.7282 ± 0.0686	0.7985 ± 0.0153	0.8726 ± 0.0267	0.7204 ± 0.0055	**0.9394 ± 0.0005**	0.7903 ± 0.0307	0.7474 ± 0.0214
LastFM	0.8057 ± 0.0424	0.7956 ± 0.0631	0.8261 ± 0.0145	0.5454 ± 0.0094	0.9225 ± 0.0009	0.8842 ± 0.0	**0.9235 ± 0.0028**
Enron	0.7640 ± 0.0310	0.6883 ± 0.0635	0.7982 ± 0.0237	0.5661 ± 0.0134	0.916 ± 0.001	0.8940 ± 0.0025	**0.9308 ± 0.0085**
SocialEvo	0.8527 ± 0.0303	0.8954 ± 0.0034	0.8944 ± 0.0102	0.6497 ± 0.004	**0.9118 ± 0.0003**	—	0.8682 ± 0.0324
UCI	0.7298 ± 0.0152	0.4606 ± 0.0209	0.8306 ± 0.0177	0.704 ± 0.0046	0.9421 ± 0.0012	**0.9720 ± 0.0024**	0.9658 ± 0.0125
CollegeMsg	0.4960 ± 0.0193	0.4858 ± 0.0051	0.7983 ± 0.049	0.7184 ± 0.0014	0.941 ± 0.0026	**0.9762 ± 0.0**	0.9642 ± 0.0124
CanParl	0.5148 ± 0.0119	0.5365 ± 0.0064	0.5596 ± 0.0141	0.5814 ± 0.0041	0.6915 ± 0.0578	**0.8469 ± 0.0161**	0.6058 ± 0.0812
Contact	0.9162 ± 0.0051	0.8334 ± 0.0620	0.9411 ± 0.0071	0.5922 ± 0.0056	0.9688 ± 0.0023	**0.9762 ± 0.0**	0.9489 ± 0.0091
Flights	0.9190 ± 0.0081	0.8707 ± 0.0121	0.9439 ± 0.0043	0.8361 ± 0.0039	**0.9834 ± 0.0002**	0.9201 ± 0.0	0.9817 ± 0.0026
UNTrade	0.6392 ± 0.0132	0.6232 ± 0.0188	0.5603 ± 0.0106	*	**0.7361 ± 0.0009**	0.5640 ± 0.0137	0.6586 ± 0.0543
USLegis	0.5557 ± 0.0107	0.5687 ± 0.0008	0.6048 ± 0.0047	0.5637 ± 0.0048	0.9694 ± 0.0028	**0.971 ± 0.0009**	0.6946 ± 0.0198
UNVote	0.5242 ± 0.0050	0.5118 ± 0.0037	0.5702 ± 0.0099	0.5204 ± 0.004	0.6014 ± 0.0013	0.6025 ± 0.0	**0.7637 ± 0.0023**
Taobao	0.6696 ± 0.0025	0.6717 ± 0.0006	0.6761 ± 0.0015	0.5293 ± 0.0096	0.7389 ± 0.0026	0.8815 ± 0.0045	**0.9992 ± 0.0001**

Inductive New-Old							
Dataset \ Model	JODIE	DyRep	TGN	TGAT	CAWN	NeurTW	NAT
Reddit	0.9399 ± 0.0112	0.9552 ± 0.0027	0.9749 ± 0.0006	0.9659 ± 0.0004	0.9871 ± 0.0003	0.9810 ± 0.0015	**0.9947 ± 0.0014**
Wikipedia	0.9127 ± 0.0078	0.8947 ± 0.0040	0.9724 ± 0.0008	0.9223 ± 0.0021	0.9901 ± 0.0002	0.9884 ± 0.0007	**0.9959 ± 0.0018**
MOOC	0.7366 ± 0.5977	0.8011 ± 0.0092	0.8669 ± 0.0328	0.7263 ± 0.0059	**0.9408 ± 0.0022**	0.7907 ± 0.0336	0.7677 ± 0.0175
LastFM	0.7448 ± 0.0034	0.7024 ± 0.0532	0.7661 ± 0.0232	0.5447 ± 0.0023	0.8906 ± 0.0021	0.835 ± 0.0	**0.9235 ± 0.001**
Enron	0.7526 ± 0.0158	0.6742 ± 0.0632	0.7918 ± 0.0209	0.5729 ± 0.0189	0.9168 ± 0.0061	0.8925 ± 0.0090	**0.9319 ± 0.0092**
SocialEvo	0.8521 ± 0.0403	**0.8999 ± 0.0046**	0.8972 ± 0.0107	0.6578 ± 0.0041	0.8830 ± 0.0008	—	0.8437 ± 0.0569
UCI	0.6891 ± 0.0166	0.4574 ± 0.0207	0.8243 ± 0.0205	0.6826 ± 0.0084	0.9414 ± 0.002	0.9732 ± 0.0040	**0.9768 ± 0.0127**
CollegeMsg	0.5000 ± 0.0227	0.4834 ± 0.0177	0.7954 ± 0.0349	0.701 ± 0.0058	0.9407 ± 0.0017	0.9719 ± 0.0014	**0.9763 ± 0.0133**
CanParl	0.5143 ± 0.0043	0.5168 ± 0.0170	0.552 ± 0.0135	0.574 ± 0.0054	0.6952 ± 0.0518	**0.8417 ± 0.0132**	0.6027 ± 0.0787
Contact	0.9150 ± 0.0058	0.8253 ± 0.0637	0.9421 ± 0.0055	0.5915 ± 0.0049	0.9689 ± 0.0029	**0.9757 ± 0.0**	0.9384 ± 0.0175
Flights	0.9128 ± 0.0095	0.8657 ± 0.0117	0.9412 ± 0.0039	0.833 ± 0.0031	0.9827 ± 0.0002	0.9161 ± 0.0	**0.9845 ± 0.0033**
UNTrade	0.6333 ± 0.0102	0.6101 ± 0.0196	0.5622 ± 0.014	*	**0.7375 ± 0.001**	0.5692 ± 0.0185	0.5844 ± 0.053
USLegis	0.5567 ± 0.0106	0.5490 ± 0.0143	0.5651 ± 0.0131	0.5695 ± 0.0099	**0.9703 ± 0.0027**	0.9671 ± 0.0027	0.5024 ± 0.0511
UNVote	0.5348 ± 0.0072	0.5126 ± 0.0103	0.5724 ± 0.0107	0.5196 ± 0.0022	0.6050 ± 0.0019	0.6036 ± 0.0	**0.7598 ± 0.0167**
Taobao	0.6838 ± 0.0045	0.6884 ± 0.0013	0.6944 ± 0.0038	0.5309 ± 0.0189	0.7374 ± 0.0032	0.8687 ± 0.0010	**0.9997 ± 0.0001**

Inductive New-New							
Dataset \ Model	JODIE	DyRep	TGN	TGAT	CAWN	NeurTW	NAT
Reddit	0.9199 ± 0.0167	0.9384 ± 0.0064	0.9727 ± 0.0004	0.9523 ± 0.0056	**0.9958 ± 0.0017**	0.9890 ± 0.0003	0.9951 ± 0.0005
Wikipedia	0.9307 ± 0.0060	0.9329 ± 0.0028	0.9822 ± 0.0009	0.9592 ± 0.0039	0.9941 ± 0.0004	0.9963 ± 0.0003	**0.9979 ± 0.0009**
MOOC	0.6623 ± 0.0189	0.7135 ± 0.0148	0.8651 ± 0.0059	0.7239 ± 0.0052	**0.935 ± 0.0009**	0.7871 ± 0.0221	0.6654 ± 0.0155
LastFM	0.8558 ± 0.0110	0.8388 ± 0.0209	0.8121 ± 0.0046	0.536 ± 0.0217	0.9716 ± 0.0008	0.9585 ± 0.0	**0.9722 ± 0.0013**
Enron	0.6525 ± 0.0146	0.6312 ± 0.0449	0.7391 ± 0.0196	0.538 ± 0.0093	**0.9556 ± 0.0055**	0.9358 ± 0.0008	0.9503 ± 0.0079
SocialEvo	0.5958 ± 0.0391	0.7312 ± 0.0103	0.8268 ± 0.003	0.5096 ± 0.0097	**0.9150 ± 0.0013**	—	0.9112 ± 0.0563
UCI	0.6249 ± 0.0198	0.5062 ± 0.0032	0.8393 ± 0.0155	0.7758 ± 0.0033	0.9488 ± 0.0012	**0.9736 ± 0.0008**	0.9518 ± 0.0211
CollegeMsg	0.5212 ± 0.0244	0.5328 ± 0.0117	0.8244 ± 0.0098	0.7929 ± 0.0029	0.9484 ± 0.0039	**0.9797 ± 0.0008**	0.95 ± 0.0257
CanParl	0.4697 ± 0.0043	0.4794 ± 0.0057	0.5553 ± 0.0258	0.6004 ± 0.0087	0.6671 ± 0.0795	**0.8511 ± 0.0079**	0.5989 ± 0.0571
Contact	0.7381 ± 0.0145	0.6601 ± 0.0432	0.8916 ± 0.0075	0.5779 ± 0.0044	0.9670 ± 0.0031	**0.9704 ± 0.0**	0.9535 ± 0.0044
Flights	0.9250 ± 0.0065	0.6312 ± 0.0449	0.9644 ± 0.0015	0.8608 ± 0.0049	0.9882 ± 0.0009	0.9496 ± 0.0	**0.9906 ± 0.0009**
UNTrade	0.5801 ± 0.0112	0.5344 ± 0.0130	0.5164 ± 0.0056	*	**0.7404 ± 0.0023**	0.5685 ± 0.0298	0.6785 ± 0.0289
USLegis	0.5250 ± 0.0045	0.5523 ± 0.0127	0.5582 ± 0.02	0.5434 ± 0.0203	0.9767 ± 0.0055	**0.9803 ± 0.0005**	0.8627 ± 0.0196
UNVote	0.4973 ± 0.0145	0.4856 ± 0.0078	0.5502 ± 0.0096	0.5337 ± 0.0046	0.5830 ± 0.0076	0.5964 ± 0.0	**0.7549 ± 0.035**
Taobao	0.6764 ± 0.0013	0.676 ± 0.0011	0.6739 ± 0.0016	0.5222 ± 0.0033	0.7390 ± 0.0147	0.9025 ± 0.0035	**0.9997 ± 0.0001**

Table 6: GPU utilization of models on link prediction task. "*" denotes that TGAT layer cannot find suitable neighbors within given time interval and encounters error. The best and second-best results are highlighted as **bold red** and underlined blue.

Model / Dataset	JODIE	DyRep	TGN	TGAT	CAWN	NeurTW	NAT
	GPU Utilization (%)						
Reddit	21	22	**41**	39	26	31	40
Wikipedia	34	**46**	36	35	28	17	38
MOOC	14	29	35	35	18	14	**45**
LastFM	22	28	38	22	23	22	**48**
Enron	18	24	41	24	25	22	**51**
SocialEvo	24	25	42	25	17	22	**46**
UCI	30	25	35	33	27	**58**	44
CollegeMsg	21	32	46	47	25	34	**48**
CanParl	26	27	**54**	47	24	22	51
Contact	25	22	40	29	19	22	**50**
Flights	20	20	26	35	18	24	**43**
UNTrade	19	23	50	*	15	23	**53**
USLegis	22	28	54	38	26	**55**	53
UNVote	12	22	37	35	16	46	**54**
Taobao	29	**56**	31	55	22	53	38

Table 7: Model efficiency on the node classification task. We report seconds per epoch as **Runtime**, the averaged number of epochs for convergence before early stopping as **Epoch**, the maximum RAM usage as **RAM**, the maximum GPU memory usage as **GPU Memory**, and the maximum GPU utilization usage as **GPU Utilization**, respectively. "x" indicates that the model cannot converge within 48 hours. The best and second-best results are highlighted as **bold red** and underlined blue.

	Runtime (second)							**Epoch**						
Model / Dataset	JODIE	DyRep	TGN	TGAT	CAWN	NeurTW	NAT	JODIE	DyRep	TGN	TGAT	CAWN	NeurTW	NAT
Reddit	65.98	75.07	73.99	**16.99**	1,913.83	24,016.13	28.24	6	7	8	**4**	x	x	6
Wikipedia	15.75	14.55	14.58	**4.95**	351.54	2,723.37	6.39	4	14	**3**	7	6	4	**3**
MOOC	28.57	32.91	27.95	**8.51**	1,146.76	7,466.04	17.55	8	**3**	5	7	10	x	**3**
	RAM (GB)							**GPU Memory** (GB)						
Reddit	3.8	3.4	**3.3**	4.1	45.7	16.7	**3.3**	2.9	**2.8**	3.1	3.0	3.7	3.0	3.0
Wikipedia	**2.6**	**2.6**	**2.6**	3.1	8.2	8.2	**2.6**	2.6	2.4	2.8	2.5	3.1	2.5	**1.7**
MOOC	2.9	2.9	2.9	3.1	45.1	32.6	**2.5**	1.9	1.8	1.9	2.1	2.3	1.7	**1.3**
	GPU Utilization (%)													
Reddit	41	40	42	38	18	36	**52**							
Wikipedia	37	30	41	31	25	**80**	48							
MOOC	35	33	40	30	13	56	**62**							

D.2 GPU Utilization Comparison for Link Prediction

In Table 6, we report the GPU utilization results on the link prediction task. NAT obtains the best or second-best results on 13 datasets out of 15, followed by TGN (9 out of 15). The data structure, called *N-cache*, designed in NAT supports parallel access and updates of *dictionary-type neighborhood representation* on GPUs. Therefore, NAT achieves the best performance regarding GPU utilization. TGN proposes a highly efficient parallel processing strategy to handle temporal graph, so that TGN has the second-best performance on GPU utilization.

D.3 Efficiency Comparison of Node Classification Task

We compare the efficiency results on node classification task and show the results in Table 7. Regarding runtime per epoch, TGAT achieves the fastest performance on all three datasets, followed by NAT. Similar to the runtime results on the link prediction task, the training process of CAWN and NeurTW is much slower due to the inefficient temporal walk. As for the averaged number of epochs for convergence, NAT ranks top-2 on all three datasets, followed by JODIE (2 out of 3), TGN (2 out of 3). RAM results reveal that CAWN and NeurTW consume much more memory due to the temporal walk and complex sampling strategy. Most models need 1 - 3 GB of GPU memory, similar

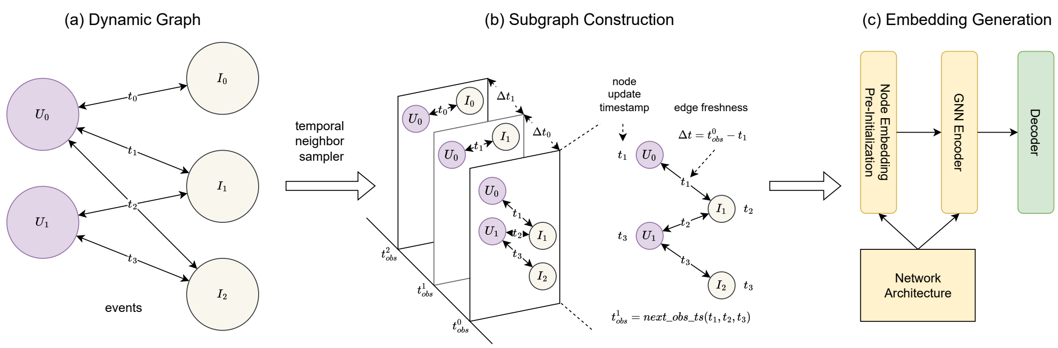


Figure 2: Workflow of TeMP. U denotes user, and I denotes item.

to the link prediction task. Due to the parallel access and update of representation on GPU, NAT achieves the highest GPU utilization.

E Results of TeMP

Upon the anatomy of the existing methods, we propose a novel temporal graph neural network, called TeMP. As shown in Figure 2, given a dynamic graph (a), the processing of TeMP is as follows:

- *Subgraph Construction (b).* We construct a subgraph with a temporal neighbor sampler with intervals adaptive to data. We try to find a reference timestamp and sample a subgraph before this timestamp. We have conducted experiments at various quantiles, and chosen the mean timestamp since it obtains the overall best performance.
- *Embedding Generation (c).* Upon subgraph construction, TeMP generates temporal embeddings for nodes and edges. The model architecture consists of three main components: temporal label propagation (LPA), message-passing operators, and a sequence updater. The temporal LPA captures the motif pattern, while the message-passing operators aggregate the original edge features. The sequence updater chooses RNN to update the embeddings with a memory module. Furthermore, TeMP uses a pre-initialization strategy to generate initial temporal node embeddings.

E.1 Experimental Results

Link Prediction. The AUC and AP results of TeMP on link prediction task are presented in Table 8, and the efficiency results are shown in Table 9. TeMP performs relatively well in the transductive setting, while lags behind CAWN, NeurTW, and NAT. Due to efficient subgraph sampling and parallel dataloader, TeMP outperforms other baselines regarding GPU memory and GPU utilization.

Node Classification. The experimental results of TeMP on node classification task are presented in Table 10. TeMP achieves the best AUC on Wikipedia dataset and the second-best AUC on Reddit dataset, demonstrating that TeMP can effectively capture temporal evolution of nodes. Similarly, TeMP consumes relatively low GPU memory and can better utilize the computation power of GPU.

Table 8: AUC and AP results of TeMP on link prediction task. We highlight the numbers as **bold red** and underlined blue if TeMP achieves the best and second-best results compared to the TGNN models in the main paper.

	AUC			
	Transductive	Inductive	Inductive New-Old	Inductive New-New
Reddit	**0.99 ± 0.0001**	0.9843 ± 0.0002	0.9818 ± 0.0001	0.9843 ± 0.0003
Wikipedia	0.9801 ± 0.0005	0.9669 ± 0.0008	0.9504 ± 0.0005	0.97 ± 0.0013
MOOC	0.8249 ± 0.0027	0.8277 ± 0.0036	0.832 ± 0.0031	0.7621 ± 0.0086
LastFM	0.865 ± 0.0012	0.8879 ± 0.0015	0.8333 ± 0.0014	0.9228 ± 0.0007
Enron	0.8717 ± 0.0122	0.8485 ± 0.0153	0.8491 ± 0.009	0.7798 ± 0.0329
SocialEvo	0.9303 ± 0.0005	0.9039 ± 0.0005	0.9017 ± 0.0013	0.8485 ± 0.0041
UCI	0.8925 ± 0.001	0.7955 ± 0.0054	0.7787 ± 0.0032	0.7318 ± 0.0017
CollegeMsg	0.8873 ± 0.0018	0.8027 ± 0.0029	0.7848 ± 0.0022	0.7342 ± 0.0047
CanParl	0.7801 ± 0.0139	0.5414 ± 0.0176	0.5313 ± 0.0253	0.5683 ± 0.0293
Contact	0.958 ± 0.0013	0.9474 ± 0.0015	0.9474 ± 0.0011	0.7835 ± 0.0041
Flights	**0.987 ± 0.0003**	0.9708 ± 0.0006	0.9692 ± 0.0006	0.973 ± 0.0009
UNTrade	0.6011 ± 0.0009	0.5732 ± 0.0006	0.5703 ± 0.0006	0.5539 ± 0.0008
USLegis	0.7073 ± 0.0114	0.5493 ± 0.0183	0.5609 ± 0.0123	0.5425 ± 0.0234
UNVote	0.5571 ± 0.0053	0.5419 ± 0.006	0.5409 ± 0.0066	0.5696 ± 0.0038
Taobao	**0.939 ± 0.0005**	0.8513 ± 0.0018	0.8249 ± 0.003	0.8502 ± 0.0025
	AP			
	Transductive	Inductive	Inductive New-Old	Inductive New-New
Reddit	**0.9904 ± 0.0001**	0.9849 ± 0.0002	0.9828 ± 0.0002	0.9757 ± 0.0007
Wikipedia	0.9817 ± 0.0004	0.9696 ± 0.0004	0.9562 ± 0.0005	0.9666 ± 0.0016
MOOC	0.7935 ± 0.0033	0.7918 ± 0.0039	0.7989 ± 0.0035	0.7328 ± 0.0089
LastFM	0.8717 ± 0.0014	0.8952 ± 0.0015	0.8468 ± 0.0015	0.8984 ± 0.0014
Enron	0.85 ± 0.0141	0.8274 ± 0.0167	0.8324 ± 0.0104	0.757 ± 0.0307
SocialEvo	0.9098 ± 0.0005	0.8767 ± 0.0004	0.8752 ± 0.0021	0.7907 ± 0.0056
UCI	0.8968 ± 0.0007	0.8104 ± 0.0054	0.7969 ± 0.0047	0.7594 ± 0.0047
CollegeMsg	0.8928 ± 0.0026	0.8206 ± 0.0041	0.8018 ± 0.003	0.7505 ± 0.0045
CanParl	0.6871 ± 0.015	0.5388 ± 0.0074	0.5341 ± 0.0037	0.5554 ± 0.015
Contact	0.9525 ± 0.0016	0.9429 ± 0.0019	0.944 ± 0.0012	0.7992 ± 0.0031
Flights	0.9857 ± 0.0003	0.9687 ± 0.0005	0.9661 ± 0.0006	0.9731 ± 0.001
UNTrade	0.5855 ± 0.0007	0.564 ± 0.0007	0.5593 ± 0.0011	0.5634 ± 0.0008
USLegis	0.6493 ± 0.0078	0.529 ± 0.01	0.5519 ± 0.0088	0.5375 ± 0.0218
UNVote	0.5402 ± 0.0038	0.536 ± 0.005	0.5354 ± 0.0041	0.5481 ± 0.0052
Taobao	**0.9385 ± 0.0004**	0.8493 ± 0.0033	0.8243 ± 0.0049	0.8448 ± 0.0056

Table 9: Efficiency of TeMP on link prediction task.

	Efficiency				
	Runtime (second)	Epoch	RAM (GB)	GPU Memory (GB)	GPU Utilization (%)
Reddit	304.84	27	5.4	2.3	**86**
Wikipedia	51.00	22	3.4	1.4	**71**
MOOC	100.64	23	2.9	**1.3**	**57**
LastFM	471.25	48	3.6	1.4	**57**
Enron	38.68	20	2.8	**1.1**	31
SocialEvo	670.08	30	3.77	**1.1**	30
UCI	43.98	26	2.9	**1.4**	**50**
CollegeMsg	11.55	30	3	**1.2**	**49**
CanParl	13.80	7	2.8	**1.1**	37
Contact	421.15	54	4.1	1.3	13
Flights	859.04	38	3.9	1.4	**76**
UNTrade	80.96	10	3	1.3	37
USLegis	10.31	14	2.6	**1.1**	**57**
UNVote	165.32	12	3.3	1.3	39
Taobao	17.78	16	3.1	1.5	42

Table 10: AUC and efficiency results of TeMP on node classification task.

	AUC	Efficiency				
		Runtime (second)	Epoch	RAM (GB)	GPU Memory (GB)	GPU Utilization (%)
Reddit	0.6357 ± 0.0265	170.66	7	4.4	3.0	**66**
Wikipedia	**0.8873 ± 0.0078**	24.18	13	3.5	1.4	47
MOOC	0.6958 ± 0.0017	49.90	7	3.3	1.2	42

References

[1] Reddit data dump. http://files.pushshift.io/reddit/.

[2] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[3] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1269–1278, 2019.

[4] Kdd cup 2015. https://biendata.com/competition/kddcup2015/data/.

[5] Balázs Hidasi and Domonkos Tikk. Fast als-based tensor factorization for context-aware recommendation from implicit feedback. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 67–82. Springer, 2012.

[6] Enron email dataset. http://www.cs.cmu.edu/~enron/.

[7] Jure Leskovec and Andrej Krevl. Snap datasets: Stanford large network dataset collection, 2014.

[8] Anmol Madan, Manuel Cebrian, Sai Moturu, Katayoun Farrahi, et al. Sensing the "health state" of a community. *IEEE Pervasive Computing*, 11(4):36–45, 2011.

[9] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[10] Pietro Panzarasa, Tore Opsahl, and Kathleen 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.

[11] Shenyang Huang, Yasmeen Hitti, Guillaume Rabusseau, and Reihaneh Rabbany. Laplacian change point detection for dynamic graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 349–358, 2020.

[12] Piotr Sapiezynski, Arkadiusz Stopczynski, David Dreyer Lassen, and Sune Lehmann. Interaction data from the copenhagen networks study. *Scientific Data*, 6(1):315, 2019.

[13] Farimah Poursafaei, Andy Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[14] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. Bringing up opensky: A large-scale ads-b sensor network for research. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pages 83–94. IEEE, 2014.

[15] Graham K MacDonald, Kate A Brauman, Shipeng Sun, Kimberly M Carlson, Emily S Cassidy, James S Gerber, and Paul C West. Rethinking agricultural trade relationships in an era of globalization. *BioScience*, 65(3):275–289, 2015.

[16] Erik Voeten, Anton Strezhnev, and Michael Bailey. United Nations General Assembly Voting Data, 2009. URL https://doi.org/10.7910/DVN/LEJUQZ.

[17] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1079–1088, 2018.

[18] Ming Jin, Yuan-Fang Li, and Shirui Pan. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. In *Advances in Neural Information Processing Systems*, 2022.