MERIT: Learning Multi-level Representations on Temporal Graphs

Binbin Hu[1*], **Zhengwei Wu**[1*], **Jun Zhou**[1], **Ziqi Liu**[1], **Zhigang Huangfu**[1], **Zhiqiang Zhang**[1] and **Chaochao Chen**[2†]

[1]Ant Group

[2]Zhejiang University

{bin.hbb, zejun.wzw, ziqiliu, zhigang.hfzg, lingyao.zzq, jun.zhoujun}@antfin.com

zjuccc@zju.edu.cn

Abstract

Recently, representation learning on temporal graphs has drawn increasing attention, which aims at learning temporal patterns to characterize the evolving nature of dynamic graphs in real-world applications. Despite effectiveness, these methods commonly ignore the individual- and combinatorial-level patterns derived from different types of interactions (*e.g.,* user-item), which are at the heart of the representation learning on temporal graphs. To fill this gap, we propose MERIT, a novel multi-level graph attention network for inductive representation learning on temporal graphs. We adaptively embed the original timestamps to a higher, continuous dimensional space for learning individual-level periodicity through Personalized Time Encoding (PTE) module. Furthermore, we equip MERIT with Continuous time and Context aware Attention (Coco-Attention) mechanism which chronologically locates most relevant neighbors by jointly capturing multi-level context on temporal graphs. Finally, MERIT performs multiple aggregations and propagations to explore and exploit high-order structural information for downstream tasks. Extensive experiments on four public datasets demonstrate the effectiveness of MERIT on both (inductive/transductive) link prediction and node classification task.

1 Introduction

Graph representation learning, which is devoted to embedding graph into low dimensional space, has drawn increasing attention in recent years. Following this line, several efforts have been made for graph structural feature extraction and attained considerable success, most notably proximity-preserving methods [Grover and Leskovec, 2016; Wang *et al.*, 2016; Zhang *et al.*, 2018] and recently emerging graph neural network (GNN) based methods [Kipf and Welling, 2017; Velickovic *et al.*, 2018; Hamilton *et al.*, 2017]. Despite excellent performance, these methods only consider static or non-temporal graphs, while real-world graphs are dynamic and continuously evolving, such as social graphs and user-item interaction graphs derived from recommender systems. Ignoring such temporal information may achieve unpromising performance for dynamic graph learning. Taking Fig. 1 as an example, the target user clicks different items over a period of time, indicating his/her intent or interest changes over time. This phenomenon drives us to learn the dynamic representation for the target user to adapt for the evolution of the graph structure.

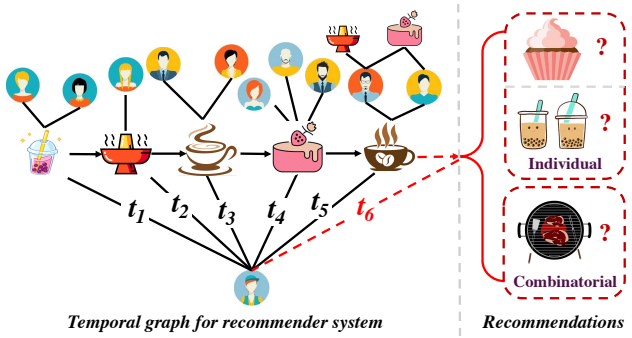


Figure 1: A toy example of temporal graph for recommender system. Different interactions are predicted when individual- and combinatorial-level context is separately considered.

To leverage temporal information, a series of recent research has shifted increasing attention towards representation learning on temporal graphs based on temporal random walk [Du *et al.*, 2018; Singer *et al.*, 2019], snapshots [Yu *et al.*, 2018; Guo *et al.*, 2019; Yang *et al.*, 2020] and temporal kernels over continuous time [Zuo *et al.*, 2018; Lu *et al.*, 2019; Huang *et al.*, 2020; Xu *et al.*, 2020]. Although considerable success has been attained, current studies commonly ignore the fact that different types of interactions (*e.g.,* user-item) on temporal graphs show temporal patterns in multi-level, which potentially implies users' underlying preferences [1], as shown in Fig. 1. Additionally, most of them overlook rich attributes on nodes and edges, and usually are designed for specific tasks (*e.g.,* traffic flow forecasting and financial risk analysis). We believe it is critically

[*]Equal contributions.

[†]Corresponding author.

[1]Multi-level context also arises in other general graphs, *e.g.,* temporal financial graphs and social graphs.

important to develop an inductive architecture for temporal graph representation in a more principled way, which hinges on jointly characterizing individual- and combinatorial-level patterns:

- **Individual level.** Due to evolving nature on temporal graphs, we are interested in patterns related to timespans between each individual node involved, where different time intervals may imply different correlations between adjacent interactions. Taking Fig. 1 as an example, the target user at t_6 is willing to interact with "Dessert" when $|t_4 - t_3|$ is small (*e.g.,* related necessities) while "Milky tea" is preferred when $|t_4 - t_3|$ and $|t_5 - t_4|$ are large (*e.g.,* daily / weekly intents).
- **Combinatorial level.** Current graph attention based methods [Velickovic *et al.*, 2018; Xu *et al.*, 2020] independently model each neighbor's effect on the target node, but ignore the effect from the combinatorial context, which aims at capturing high-level semantic among neighbors in continuous time. Specifically, given $\langle Milky\ tea; t_1\rangle$ and $\langle Hot\ pot; t_2\rangle$ in Fig. 1 (Assuming that $|t_2 - t_1| < 12h$, and $|t_3 - t_2|; |t_4 - t_3|; |t_5 - t_4| > 12h$), we consider their effects towards the target node from following two aspects. If we consider them independently, "Dessert" or "Milky tea" will be recommended based on the consumption habit in daily life, since the target user shows similar preferences at $t_3; t_4; t_5$. On the comparison, if we jointly consider them as the combinatorial-level context, we could abstract that the target user maybe in shopping mall, a high-level semantic hard to be captured in the first way. Therefore, "Barbecue" is more proper to be recommended. In sum, such context, as high-level semantic among neighbors, is of crucial importance for temporal graph representation learning.

In this paper, by integrating above properties together, we propose MERIT, a novel Multi-lEvel gRaph attentIon neTwork for inductive representation learning on temporal graphs. In particular, we take the inspiration from the recently emerging graph neural networks, which have the potential of capturing high-order structural information in an inductive manner [Cai *et al.*, 2018; Velickovic *et al.*, 2018; Hamilton *et al.*, 2017], but have not been explored much for temporal graphs in continuous time. Benefiting from Mercer's Theorem [Minh *et al.*, 2006; Xu *et al.*, 2019], we propose a Personalized Time Encoding (PTE) module for transforming the original timespans into a higher, continuous space in a personalized manner, targeting for effectively preserving individual-level periodicity on temporal graphs. Subsequently, a novel Continuous time and context aware Attention (Coco-Attention) mechanism is developed to chronologically emphasize core local structure by jointly considering individual- and combinatorial-level context. At last, MERIT performs multiple aggregations and propagations to comprehensively explore high-order structural information in continuous time, followed by the end-to-end training for various downstream tasks. In sum, we make the following contributions:

- We highlight the crucial importance of explicitly exploring individual- and combinatorial-level patterns for representation learning on temporal graphs, and make the first step to characterize these properties together with an unified model in a principled way.
- We propose a novel model MERIT that is equipped with PTE module for depicting individual-level periodicity in a personalized manner and Coco-Attention mechanism where multi-level context is jointly captured.
- We perform extensive experiments on four public datasets in both transductive and inductive settings. Results demonstrate that MERIT consistently and significantly outperforms various state-of-the-art methods.

2 Related Work

Graph representation learning has shown its potential in structure feature extraction and has been widely applied in many data mining tasks [Cai *et al.*, 2018]. Conventional graph representation learning methods have been developed to preserve graph topology [Grover and Leskovec, 2016; Wang *et al.*, 2016; Zhang *et al.*, 2018]. With the advent of deep learning methods, significant efforts have been devoted to developing neural network-based representation learning methods [Kipf and Welling, 2017; Hamilton *et al.*, 2017; Velickovic *et al.*, 2018; Liu *et al.*, 2022; Bo *et al.*, 2022]. Unfortunately, they are only designed for static graphs, which cannot capture dynamic natures on temporal graphs.

Recently, attention is increasingly shifted towards representation learning on temporal graphs [Kazemi *et al.*, 2020]. Early methods [Du *et al.*, 2018; Singer *et al.*, 2019; Nguyen *et al.*, 2018] attempt to learn dynamic representations with temporal random walk and extended skip-gram model, and subsequently achieve superior performance on transductive tasks. Taking advantages of information propagation, a series of spatial-temporal graph neural networks [Seo *et al.*, 2018; Yang *et al.*, 2020; Guo *et al.*, 2019] are proposed for temporal graphs, which utilize the temporal aggregator for integrating representations learned from each graph snapshot. As a major weakness, these methods ignore the dynamic nature in each graph snapshot, resulting in sub-optimal performance. In order to capture the evolving patterns of graph representation over time, a series of works are proposed to embed temporal graph in continuous time [Kumar *et al.*, 2019; Lu *et al.*, 2019; Zuo *et al.*, 2018; Xu *et al.*, 2020; Huang *et al.*, 2020; Wang *et al.*, 2021].

Nevertheless, all the above-mentioned methods deal with temporal graphs by independently modeling each dynamic interaction, failing to explore the effectiveness of multi-level patterns. In this paper, we jointly incorporate individual- and combinatorial-level context into an unified graph neural network framework for powerful inductive representation learning on temporal graphs.

3 The Proposed Method

In this section, we propose **MERIT**, a Multi-lEvel gRaph attentIon neTwork for representation learning on temporal graphs. The overall architecture of the proposed MERIT is shown in Fig. 2. Let's begin with some notations and definitions.

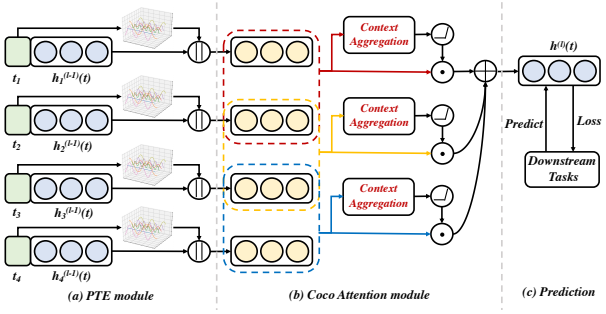


Figure 2: Overview of the proposed MERIT model. (a) Mapping timestamp to the continuous differentiable vector space in a personalized way with PTE module. (b) Aggregating important and relevant neighbors by jointly considering multi-level context with Coco Attention module . (c) Loss calculation and end-to-end optimization for the specific downstream task.

3.1 Notations and Definitions

Our study focuses on temporal graphs, which can be defined as follows,

Definition 1. ***Temporal Graph.*** *A temporal graph* $\mathcal{G} = \{\mathcal{V}; \mathcal{E}; \mathcal{T}\}$ *is a form of graph with timestamps, where* $\mathcal{V}$ *and* $\mathcal{E}$ *are sets of nodes and edges, and* $\mathcal{T}$ *is the sequence of timestamps. Each edge* $(u; v) \in \mathcal{E}$ *is associated with a timestamp* $t \in \mathcal{T}$*, referring to an interaction involving node* u *and node* v *at time* t*.*

Now, we formally define the problem studied in this paper as follows,

Definition 2. ***Temporal Graph Representation Learning.*** *Given a temporal graph* $\mathcal{G} = \{\mathcal{V}; \mathcal{E}; \mathcal{T}\}$*, we aim to learn a mapping function* $f : \mathcal{V} \times \mathcal{T} \to \mathbb{R}^d$*, where* d *is the number of embedding dimensions and* $d \ll |\mathcal{V}|$*. The mapping function* f *is expected to produce representation for each target node* $v \in \mathcal{V}$ *at target timestamp* $t \in \mathcal{T}$ *with corresponding sub-graph before timestamp* t*.*

3.2 Personalized Time Encoding

Time encoding is at the heart of the representation learning on temporal graphs, which refers to capturing dynamic nature over a period of time. Most importantly, we should notice that temporal modeling is highly time-sensitive and each node plays a different role and shows different pattern towards neighbors. Hence, we introduce a Personalized Time Encoding (PTE) module that can adaptively learn individual-level periodicity for each target node.

Formally, the goal of time encoding is to find a mapping $: \mathcal{T} \to \mathcal{R}^d$ from time domain $\mathcal{T}$ to d-dimensional vector space. Considering arbitrary timestamp $t \in \mathcal{T}$, suggested by Mercer's Theorem [Minh *et al.*, 2006; Xu *et al.*, 2019], we define the mapping as follows:

$$t \mapsto \quad (t) := [\sqrt{c_1}\ _1(t); \sqrt{c_2}\ _2(t); :::]; \tag{1}$$

where $\{c_i\}_{i=1}^{1}$ and $\{\ _i(\cdot)\}_{i=1}^{1}$ is an associated set of non-negative eigenvalues and a sequence of eigenfunctions, respectively [Minh *et al.*, 2006]. Intuitively, temporal patterns can be detected from a finite set of periodic kernels. Following the proposition introduced in [Xu *et al.*, 2019], we further formulate the mapping function with frequency parameter $!$ as below:

$$t \mapsto \quad _!(t) := [\sqrt{c_1}; :::; \sqrt{c_{2j}}cos(\frac{j\ t}{!}); \sqrt{c_{2j+1}}sin(\frac{j\ t}{!}); :::]: \tag{2}$$

Fortunately, such a Fourier series-like form has nice truncation properties, which drive us to truncate above mapping function $_!(t)$ as $_{!;d}(t)$. Subsequently, by concatenating multiple truncated periodic mapping functions, parameterized by the frequency set $\{!_1; :::; !_k\}$, we encode the functional time as:

$$t \mapsto \quad _d(t) := [\ _{!_1;d}(t)||:::||\ _{!_k;d}(t)]^T: \tag{3}$$

It is worthwhile to note that the Fourier coefficients in Eq. 2 (*i.e.,* $c_i; i = 1; 2; :::$) are not personalized. That is, the above time encoding is irrelevant to the corresponding node representations on temporal graphs. Specifically, Eq. 3 tends to obtain the same time encoding for two different nodes with same timestamps. Such a paradigm for time encoding is improper to be directly incorporated into following attention mechanism. Therefore, giving a target node u, we further formulate our personalized periodic kernel for u as follows,

$$\begin{aligned} u; t \mapsto \quad _!(u; t) :=& [\sqrt{c_1(u)}; :::; \sqrt{c_{2j}(u)}cos(\frac{j\ t}{!}); \\ & \sqrt{c_{2j+1}(u)}sin(\frac{j\ t}{!}); :::]: \end{aligned} \tag{4}$$

Here, $c_i(u) : \mathbb{R}^d \to \mathbb{R}; i = 1; 2; ; :::$ are personalized mapping functions for Fourier coefficients *w.r.t.* node u. Empirically, we implement $c_i(\cdot)$ as a multi-layer perceptron (MLP) due to its strong ability in modeling complex interaction. Moreover, we enforce the outputs of MLPs to be non-negative (*i.e.,* Softplus activation function is adopted in the output layer.), in order to satisfy the intrinsic properties of the Mercer's Theorem. The input of the mapping function is the representation of node u, denoted as $\mathbf{h}_u$. In sum, we redefine the time encoding via our proposed PTE module as:

$$u; t \mapsto \quad _d(u; t) := [\ _{!_1;d}(u; t)||:::||\ _{!_k;d}(u; t)]^T: \tag{5}$$

3.3 Continuous Time and Context aware Attention Mechanism

With the help of the PTE module, we encode the timestamps into a continuous, higher dimensional space for preserving the temporal patterns for each node in a personalized manner. Next, we introduce the Continuous time and Context aware Attention (Coco-Attention) mechanism that can chronologically tell the difference of local neighbors by jointly capturing multi-level context on temporal graphs.

Continuous time aware. Now, we start with the continuous time aware attention for effectively capturing local structure while preserving individual-level dynamic in temporal graphs, as well as the overall pipeline of Coco-Attention calculation for MERIT. We build upon the architecture of the recently emerging graph attention mechanism [Velickovic *et al.*, 2018] to weigh various underlying preferences for interactions between connected nodes. Specifically, given a target

node u at time t, we aim to produce the attention distribution of u towards its neighbors $\mathcal{N}_u(t) = \{v|t_v<t\}$, followed by a weighted combination for the final representations. Due to the translation-invariant property for the temporal kernel, we can alternatively use $\{t-t_v\}_{v2N_u(t)}$ as interaction times. Formally, we first calculate the Coco-Attention weight involving target node u and one of its neighbor $v \in \mathcal{N}_u(t)$ at time t as follows:

$$
\begin{aligned}
_{u;v}(t) &= \frac{\mathbf{Q}_u(t)\mathbf{K}_v^T(t)}{\sqrt{d}}; \\
\mathbf{Q}_u(t) &= [\mathbf{h}_u^{(l\ \ 1)}(t) \,||\, \mathbf{0} \,||\, \quad _d(0)] \cdot \mathbf{W}_Q; \\
\mathbf{K}_v(t) &= [\mathcal{F}^{(l\ \ 1)}(v;t;t_v) \,||\, \mathbf{e}_{u;v}(t) \,||\, \quad _d(t-t_v)] \cdot \mathbf{W}_K;
\end{aligned}
\tag{6}
$$

where "||" denotes the concatenation operation, d is the dimension of the node representation, $\mathbf{0}$ is the all-zero vector, $\mathbf{W}_Q$ and $\mathbf{W}_K$ is the projection matrice to obtain the "query" and "key" matrix, respectively [Vaswani *et al.*, 2017], and $\mathbf{e}_{u;v}(t)$ is the feature vector for the edge connecting u and v at time t. $\mathcal{F}^{(l\ \ 1)}(v;t;t_v)$ is the context modeling part, which will be introduced in next subsection. Obviously, we can easily extend it to multi-head attention for the stable training process. Due to page limitation, we omit it here.

Next, we produce the continuous time aware representation for target node u at time t through l-th layer in MERIT [2] as follows,

$$
\begin{aligned}
\mathbf{h}_u^{(l)}(t) &= \sum_{v2N_u} sof\,tmax_v(\ _{u;v}(t))\mathbf{V}_v(t); \\
\mathbf{V}_v(t) &= [\mathbf{h}_v^{(l\ \ 1)}(t) \,||\, \mathbf{e}_{u;v}(t) \,||\, \quad _d(t-t_v)] \cdot \mathbf{W}_V;
\end{aligned}
\tag{7}
$$

where $\mathbf{W}_V$ is the projection matrix to generate the "value" matrix [Vaswani *et al.*, 2017].

Context aware. As mentioned above, current Coco-Attention mechanism is sufficient to capture individual-level patterns in temporal graphs by incorporating multi-scale (small / large) time interval into attention calculation in continuous space. As another major component of Coco-Attention mechanism, we further introduce the combinatorial-level context modeling, *i.e.*, $\mathcal{F}^{(l\ \ 1)}(v;t;t_v)$. Formally, with the contextual neighbors $\mathcal{S}_{u;v}(t_v) = \{v^0 \in \mathcal{N}_u(t)|t_{v^0} \le t_v\}$ for neighbor node v at time t_v, we implement $\mathcal{F}(\cdot)$ using two types of aggregators:

- *Recursive aggregator* applies complex LSTM architecture on the sequential context to endow MERIT layer with excellent expressive capability.

$$
\mathcal{F}_R = \mathrm{LSTM}(\mathbf{h}_{v^0}^{(l\ \ 1)}(t); v^0 \in \mathcal{S}_{u;v}(t_v)): \tag{8}
$$

- *Convolutional aggregator* applies convolution-like operator for more scalable implementation.

$$
\mathcal{F}_C = \sum_{i=1}^{d\ \ 1jS_{u;v}(t_v)j} \sum_{j=0} \mathbf{h}_j^{(l\ \ 1)}(t)\mathbf{W}_{j;i}; \tag{9}
$$

where $\mathbf{W} \in \mathsf{R}^{jS_{u;v}(t_v)j\ \ d\ \ d}$ is the convolution kernel.

[2] MERIT follows the classical architectures of graph neural networks with multiple layers.

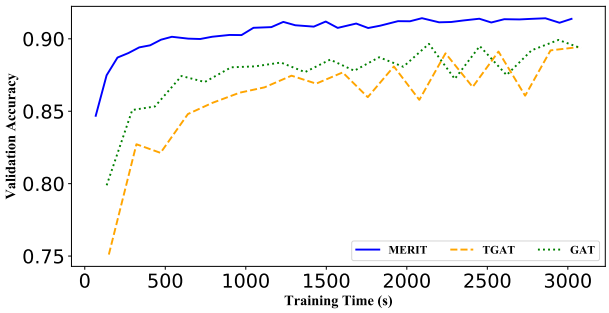


Figure 3: Analysis of running time on Wikipedia dataset.

3.4 Model Learning

Naturally, MERIT performs multiple aggregations and propagations to explore high-order structural information in a broader and deeper way. With the initial representations $\{\mathbf{h}_v^{(0)}(t)\}_{v2V}$ for nodes set $\mathcal{V}$ at time t, we rewrite the final temporal representations after L layers of MERIT as $\{\mathbf{h}_v^f(t) = \mathbf{h}_v^{(L)}(t)\}_{v2V}$. Following previous studies, we could learn parameters of MERIT through various downstream applications with specific loss functions [Xu *et al.*, 2020; Lu *et al.*, 2019], *e.g.,* link prediction and node classification.

Complexity. To speed up the training process, we randomly collect L-hop neighbors set for each node in mini-batch if we aim to stack L layers in MERIT. Since the masked self-attention operation is parallelizable [Velickovic *et al.*, 2018], the time complexity of each layer in MERIT with K heads and L layers for each batch is approximately $\mathcal{O}((K \cdot \mathcal{N})^L)$, where $\mathcal{N}$ is the number of neighbors sampled for each node. Clearly, the time complexity of MERIT is comparable with GAT, which is verified in the Fig. 3. In addition, Fig. 3 shows that MERIT achieves faster convergence speed and better performance. *Actually, MERIT has been deployed in industrial recommender systems to support very large-scale temporal graphs, consisting of hundreds of millions of nodes and edges.*

4 Experiments

In this section, we perform a series of experiments on four datasets to demonstrate the effectiveness of MERIT.

4.1 Experimental Setup

Evaluation Datasets and Metrics. We conduct extensive experiments on four widely used datasets [Kumar *et al.*, 2019] from different domains, namly **Reddit**, **Wikipedia**, **MOOC** and **LastFM**. In particular, we calculate the Average Repetitive Rate (ARR) of user behaviors for the four datasets adopted in [Bai *et al.*, 2019] to reveal the periodicity. We summarize the statistics of the four datasets in Table 1.

In our experiments, we adopt commonly used average precision (**AP**) and **Accuracy** for the link prediction evaluation and area under the ROC curve (**AUC**) for the node classification evaluation.

	Reddit	Wikipedia	MOOC	LastFM
# Nodes	11, 000	9, 227	7, 145	2, 000
# Edges	672, 477	157, 474	411, 749	1, 293, 103
# Features	172	172	4	N.A.
ARR	0.15	0.60	0.53	0.32

Table 1: Statistics of the datasets.

Baselines We compare MERIT with ten state-of-art methods, falling into three main groups: i) deep recurrent neural network based methods (*i.e.,* **Time-LSTM** [Zhu *et al.*, 2017] and **Jodie** [Kumar *et al.*, 2019]) learning dynamic embeddings from a sequence of temporal interactions, static graph neural network based methods (*i.e.,* **GraphSAGE** [Hamilton *et al.*, 2017] and **GAT** [Velickovic *et al.*, 2018]) capturing high-order structure with information propagation and temporal graph representation learning methods (*i.e.,* **CTDNE** [Nguyen *et al.*, 2018], **M^{2}DNE** [Lu *et al.*, 2019], **GCRN** [Seo *et al.*, 2018], **GraphSAGE-T**, **GAT-T** and **TGAT** [Xu *et al.*, 2020]) well designed for temporal graphs.

For our model, we report its performance with different context aggregators, *i.e.,* **MERIT**$_R$ with recursive aggregator and **MERIT**$_C$ with convolutional aggregator.

Significance Test For results in Tables 2 and 3, we use ** (or *) to indicate that the improvement of MERIT over the best performance from the best baseline is significant based on paired t-test at the significance level of 0.01 (or 0.05).

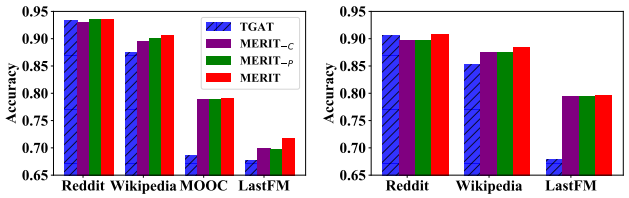
4.2 Experimental Results and Analysis

Overall Performance Comparison

From the experimental results in Table 2 and Table 3, firstly, we observe that our proposed MERIT model consistently outperforms all the baselines on both link prediction and node classification tasks, demonstrating the effectiveness of MERIT. Not surprisingly, MERIT$_R$ slightly outperforms MERIT$_C$ due to excellent expressive capability of LSTM architecture. Secondly, compared to traditional graph neural networks, the performance gains of temporal graph representation learning methods are attributed to the functional time encoding on temporal graphs. Particularly, we also observe that the performance margin in Reddit is relatively small, on account of the weak periodicity in this dataset (*i.e.,* small ARR in Table 1). Thirdly, graph neural network based methods achieve better performance than deep recurrent neural network based methods in most cases, which indicates the usefulness of high-order structural information and the rich attributes on edges.

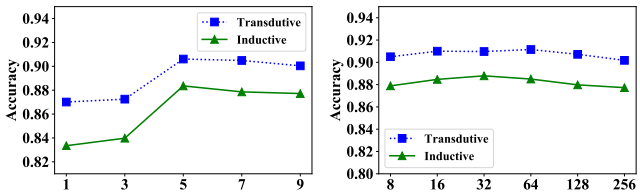
Ablation Study

We perform a series of ablation studies to better understand the traits of MERIT. Noting that MERIT will use the Recursive aggregator, *i.e.,*MERIT$_R$ by default in what follows since it performs best. Due to the page limitation, we only present results of parameter studies and attention analysis on Wikipedia dataset with link prediction task, and similar trends can be observed on other datasets.



(a) Transductive (b) Inductive

Figure 4: Impact of the personalization and context modeling.



(a) k (b) d

Figure 5: Performance comparison (Accuracy) of different k and d on Wikipedia dataset.

Impact of Personalization and Context Modeling. We prepare two variants of MERIT for comparison, namely MERIT $_P$ (MERIT without personalization modeling in PTE module) and MERIT $_C$ (MERIT without context modeling in Coco-Attention mechanism). Moreover, we select the TAGT as the reference baseline and report the comparison results in Fig. 4. Not surprisingly, we find that the overall performance order is as follows: MERIT > MERIT $_P$, MERIT $_C$ > TGAT. The results show that it is necessary to adaptively learn proper Fourier coefficients in periodic kernels for enhancing temporal representations, while related combinatorial-level context information is also of crucial importance to be captured.

Impact of the Number of Periodic Kernels. Here, we analyze the impact of the number of periodic kernels (*i.e.,* k in Eq. 5) by varying it in the set of $\{1, 3, 5, 7, 9\}$. In Fig 5 (a), it is clear that the optimal performance is achieved with $k = 5$, demonstrating a proper number of periodic kernels is beneficial to effectively capture temporal patterns. However, a larger k may lead to overfitting issue.

Impact of the Truncated Dimension. Similarly, we investigate into the impact of the truncated dimension d (See Eq. 3) by varying it in the set of $\{8, 16, 32, 64, 128, 256\}$. As shown in Fig 5 (b), MERIT achieves the best performance when $d = 32$ or $d = 64$. Overall, our model is not sensitive to this parameter due to the nice truncation properties of Fourier series-like form.

Attention Weight Analysis

We first present the macro-level analysis of the attention distributions on three datasets (*i.e.,* Reddit, Wikipedia and Mooc), which aims at analyzing how the attention weights change *w.r.t.* the timespans of previous interactions. Specifically, for each node pair $\langle u; v; t\rangle$, we plot the attention weights $\{\ _{u;v^0(t)}|v^0 \in \mathcal{N}_u(t)\} \bigcup \{\ _{u^0;v(t)}|u^0 \in \mathcal{N}_v(t)\}$

		Reddit		Wikipedia		MOOC		LastFM	
		Accuracy	AP	Accuracy	AP	Accuracy	AP	Accuracy	AP
Tr.	Time-LSTM	0.7025	0.7157	0.5625	0.5648	0.5601	0.5673	0.5103	0.5216
	Jodie	0.9088	0.9742	0.8354	0.9293	0.7822	0.7746	0.6211	0.6505
	GraphSAGE	0.9323	0.9830	0.8889	0.9599	0.7004	0.7459	0.6441	0.6922
	GAT	0.9317	0.9833	0.8807	0.9539	0.6732	0.7217	0.6548	0.6800
	CTDNE	0.7810	0.8594	0.5521	0.5689	0.5802	0.5919	0.3920	0.4399
	M^2DNE	0.8622	0.9429	0.8167	0.9091	0.6858	0.6945	0.5926	0.6201
	GCRN	0.9338	0.9829	0.8855	0.9552	0.7106	0.7462	0.6541	0.7213
	GraphSAGE-T	0.9303	0.9823	0.8969	0.9648	0.7566	0.7868	0.6791	0.7765
	GAT-T	0.9323	0.9834	0.8984	0.9647	0.7553	0.7901	0.6785	0.7576
	TGAT	0.9342	0.9837	0.8743	0.9502	0.6869	0.7157	0.6765	0.6732
	MERIT_C	0.9348	0.9840	0.9038	0.9702	0.7907	0.8521	0.7021	0.7922
	MERIT_R	**0.9355***	**0.9845***	**0.9061****	**0.9714****	**0.7908****	**0.8614****	**0.7171****	**0.8129****
	(*v.s.* best)	(**+0.14%**)	(**+0.08%**)	(**+1.02%**)	(**+0.68%**)	(**+4.52%**)	(**+9.02%**)	(**+5.59%**)	(**+4.69%**)
In.	GraphSAGE	0.9001	0.9650	0.8627	0.9442	0.6973	0.7365	-	-
	GAT	0.9018	0.9669	0.8543	0.9372	0.6610	0.6997	-	-
	GCRN	0.9002	0.9636	0.8533	0.9328	0.6942	0.7438	-	-
	GraphSAGE-T	0.8991	0.9650	0.8761	0.9566	0.7641	0.7976	-	-
	GAT-T	0.9031	0.9681	0.8803	0.9562	0.7677	0.8043	-	-
	TGAT	0.9056	0.9679	0.8536	0.9353	0.6789	0.7036	-	-
	MERIT_C	0.9064	0.9680	0.8831	0.9602	0.7944	0.8403	-	-
	MERIT_R	**0.9069***	**0.9682***	**0.8836***	**0.9605***	**0.7962****	**0.8442****	-	-
	(*v.s.* best)	(**+0.14%**)	(**+0.03%**)	(**+0.37%**)	(**+0.41%**)	(**+3.71%**)	(**+4.96%**)	(-)	(-)

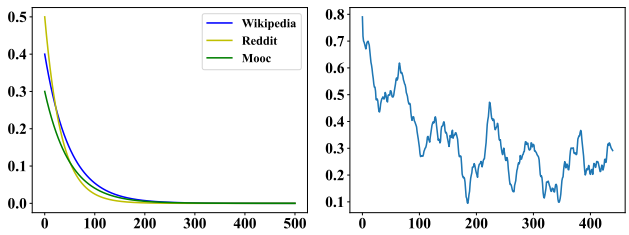
Table 2: Performance comparison on transductive (Tr.) and inductive (In.) link prediction. "-" means no feature is available on LastFM dataset so that inductive learning is unable to be performed.

	Reddit	Wikipedia	MOOC
Time-LSTM	0.6305	0.7773	0.6935
Jodie	0.6106	0.7629	0.6655
GraphSAGE	0.6502	0.7974	0.6742
GAT	0.6617	0.8480	0.6459
GCRN	0.6743	0.8575	0.6695
GraphSAGE-T	0.6594	0.8515	0.6779
GAT-T	0.6743	0.8508	0.6694
TGAT	0.6411	0.8606	0.6786
MERIT_C	0.6763	0.8694	0.6811
MERIT_R	**0.6846****	**0.8753****	**0.6881****
(*v.s.* best)	(**+5.91%**)	(**+4.08%**)	(**+5.32%**)

Table 3: Performance comparison (AUC) on node classification.

against $\{t - t_{u;v^\theta}\} \bigcup \{t - t_{u^\theta;v}\}$. we report the results in Fig. 6 (a), where a smaller timespan means a more recent interaction. It is clear that timespans and the corresponding attention weights are negatively correlated, which means that MERIT captures the temporal patterns of putting less attention on more distant interactions.

From micro-level view, we select a certain user from Wikepedia dataset and plot his/her attention weight against timespans of historical interactions as an illustrative example. In Fig. 6 (b), we observe that the attention weight decreases periodically with the increase of timespans. This finding is consistent with our intuition that user's intention or interest will periodically change over time.



(a) Macro level (b) Micro level

Figure 6: The change of attention weights *w.r.t.* timespans.

5 Conclusion and Future Work

In this paper, we proposed the novel MERIT model, which consists of PTE module for characterizing individual-level periodicity in an personalized manner and Coco-Attention mechanism for jointly capturing multi-level context. Extensive experiments demonstrate the superior performance of MERIT in both node classification and link prediction tasks. As future work, we will keep on investigating into the functional time encoding for automatically learning the parameters (*i.e.*, $\{\omega_1, \cdots, \omega_k\}$) of periodic kernels. Moreover, we will also consider incorporating temporal point process to replace current MLP component as a decoder for dynamics modeling.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No.72192823 and No.62172362).

References

[Bai *et al.*, 2019] Ting Bai, Lixin Zou, Wayne Xin Zhao, Pan Du, Weidong Liu, Jian-Yun Nie, and Ji-Rong Wen. Ctrec: A long-short demands evolution model for continuous-time recommendation. In *SIGIR*, pages 675–684, 2019.

[Bo *et al.*, 2022] Deyu Bo, BinBin Hu, Xiao Wang, Zhiqiang Zhang, Chuan Shi, and Jun Zhou. Regularizing graph neural networks via consistency-diversity graph augmentations. page AAAI, 2022.

[Cai *et al.*, 2018] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

[Du *et al.*, 2018] Lun Du, Yun Wang, Guojie Song, Zhicong Lu, and Junshan Wang. Dynamic network embedding: An extended approach for skip-gram based network embedding. In *IJCAI*, pages 2086–2092, 2018.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pages 855–864, 2016.

[Guo *et al.*, 2019] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*, pages 922–929, 2019.

[Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017.

[Huang *et al.*, 2020] Hong Huang, Zixuan Fang, Xiao Wang, Youshan Miao, and Hai Jin. Motif-preserving temporal network embedding. In *IJCAI*, pages 1237–1243, 2020.

[Kazemi *et al.*, 2020] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Kumar *et al.*, 2019] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *SIGKDD*, pages 1269–1278, 2019.

[Liu *et al.*, 2022] Hongrui Liu, Binbin Hu, Xiao Wang, Chuan Shi, Zhiqiang Zhang, and Jun Zhou. Confidence may cheat: Self-training on graph neural networks under distribution shift. page WWW, 2022.

[Lu *et al.*, 2019] Yuanfu Lu, Xiao Wang, Chuan Shi, Philip S Yu, and Yanfang Ye. Temporal network embedding with micro-and macro-dynamics. In *CIKM*, pages 469–478, 2019.

[Minh *et al.*, 2006] Ha Quang Minh, Partha Niyogi, and Yuan Yao. Mercer's theorem, feature maps, and smoothing. In *COLT*, pages 154–168, 2006.

[Nguyen *et al.*, 2018] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In *WWW*, pages 969–976, 2018.

[Seo *et al.*, 2018] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *NIPS*, pages 362–373, 2018.

[Singer *et al.*, 2019] Uriel Singer, Ido Guy, and Kira Radinsky. Node embedding over temporal graphs. In *IJCAI*, pages 4605–4612, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *SIGKDD*, pages 1225–1234, 2016.

[Wang *et al.*, 2021] Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, et al. Apan: Asynchronous propagation attention network for real-time temporal graph embedding. In *ICMD*, pages 2628–2638, 2021.

[Xu *et al.*, 2019] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Self-attention with functional time representation learning. In *NIPS*, pages 15915–15925, 2019.

[Xu *et al.*, 2020] Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *ICLR*, 2020.

[Yang *et al.*, 2020] Shuo Yang, Zhiqiang Zhang, Jun Zhou, Yang Wang, Wang Sun, Xingyu Zhong, Yanming Fang, Quan Yu, and Yuan Qi. Financial risk analysis for smes with graph-based supply chain mining. In *IJCAI*, pages 4661–4667, 2020.

[Yu *et al.*, 2018] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*, pages 3634–3640, 2018.

[Zhang *et al.*, 2018] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In *SIGKDD*, pages 2778–2786, 2018.

[Zhu *et al.*, 2017] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: Modeling user behaviors by time-lstm. In *IJCAI*, pages 3602–3608, 2017.

[Zuo *et al.*, 2018] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In *SIGKDD*, pages 2857–2866, 2018.