



# TemporalGAT: Attention-Based Dynamic Graph Representation Learning

Ahmed Fathy and Kan Li<sup>(✉)</sup>

School of Computer Science and Technology, Beijing Institute of Technology,  
Beijing 10081, China  
{ahmedfathy,likan}@bit.edu.cn

**Abstract.** Learning representations for dynamic graphs is fundamental as it supports numerous graph analytic tasks such as dynamic link prediction, node classification, and visualization. Real-world dynamic graphs are continuously evolved where new nodes and edges are introduced or removed during graph evolution. Most existing dynamic graph representation learning methods focus on modeling dynamic graphs with fixed nodes due to the complexity of modeling dynamic graphs, and therefore, cannot efficiently learn the evolutionary patterns of real-world evolving graphs. Moreover, existing methods generally model the structural information of evolving graphs separately from temporal information. This leads to the loss of important structural and temporal information that could cause the degradation of predictive performance of the model. By employing an innovative neural network architecture based on graph attention networks and temporal convolutions, our framework jointly learns graph representations contemplating evolving graph structure and temporal patterns. We propose a deep attention model to learn low-dimensional feature representations which preserves the graph structure and features among series of graph snapshots over time. Experimental results on multiple real-world dynamic graph datasets show that, our proposed method is competitive against various state-of-the-art methods.

**Keywords:** Dynamic graph representation learning · Graph attention networks · Temporal convolutional networks

## 1 Introduction

Many appealing real-world applications involve data streams that cannot be well represented in a planar structure, but exist in irregular domain. This case applies to knowledge bases [35], 3D models [18], social media [22], and biological networks [7] which are usually represented by graphs.

In graph representation learning, the key challenge is to learn a low-dimensional representation of the data that is most informative to preserve the structural information among the nodes in graphs. Through graph embedding, we can represent the nodes in a low-dimensional vector form. This paves the way

to apply machine learning in graph analysis and data mining tasks easily and efficiently such as node classification [11,22], link prediction [7], clustering [4], and visualization [30].

Recently, there has been significant interest in graph representation learning mainly focuses on static graphs [5,7,8,11,22,29] which attracted the attention of researchers due to its extensive usage in numerous real-world applications. However, a wide range of real-world applications are intrinsically dynamic and the underlying graph structure evolves over time and are usually represented as a sequence of graph snapshots over time [14].

Learning dynamic graph representations is challenging due to the time-varying nature of graph structures, where the graph nodes and edges are in continues evolution. New nodes and edges can be introduced or removed in each time step. Consequently, this requires the learned representations not only to preserve structural information of the graphs, but also to efficiently capture the temporal variations over time.

Recently, novel methods for learning dynamic graph representations have been proposed in literature. Some recent work attempts to learn dynamic graph representation such as [10,15,36,37], where they mainly apply a temporally regularized weights to enforce the smoothness of node representations from different adjacent time steps. However, these methods generally fail to learn effective representations when graph nodes exhibit substantially distinct evolutionary behaviors over time [24].

Trivedi et al. [27] handle temporal reasoning problem in multi-relational knowledge graphs through employing a recurrent neural network. However, their learned temporal representations are limited to modeling first-order proximity between nodes, while ignoring the higher-order proximities among neighborhoods which are essential for preventing the graph structure as explained in [25,34].

Recently, the authors in [24] propose dynamic graph embedding approach that leverage self-attention networks to learn node representations. This method focus on learning representations that capture structural properties and temporal evolutionary patterns over time. However, this method cannot effectively capture the structural evolution information over time, since it employs structure attention layers to each time step separately and generate node representations, which is followed by temporal attention layers to capture the variations in generated representations.

Recently, attention mechanisms have achieved great success in NLP and sequential learning tasks [1,31]. Attention mechanisms learn a function that aggregates a variable-sized inputs while focusing on the most relevant sequences of the input to make decisions, which makes them unique. An attention mechanism is commonly referred to as self-attention, when it computes the representation of a single sequence.

Veličković et al. [29] extend the self-attention mechanism and apply it on static graphs by enabling each node to attend over its neighbors. In this paper, we specifically focus on applying graph attention networks (GATs) [29] because of its effectiveness in addressing the shortcomings of prior methods based on graph

convolutions such as [8, 11]. GATs allow for assigning different weights to nodes of the same neighborhood by applying multi-head self-attention layers, which enables a leap in model capacity. Additionally, the self-attention mechanism is applied to all graph edges, and thus, it does not depend on direct access to the graph structure or its nodes, which was a limitation of many prior dynamic graph representation learning techniques.

Inspired by this recent work, we present a temporal self-attention neural network architecture to learn node representations on dynamic graphs. Specifically, we apply self-attention along structural neighborhoods over temporal dynamics through leveraging temporal convolutional network (TCN) [2, 20]. We learn dynamic node representation by considering the neighborhood in each time step during graph evolution by applying a self-attention strategy without violating the ordering of the graph snapshots.

Overall our paper makes the following contributions:

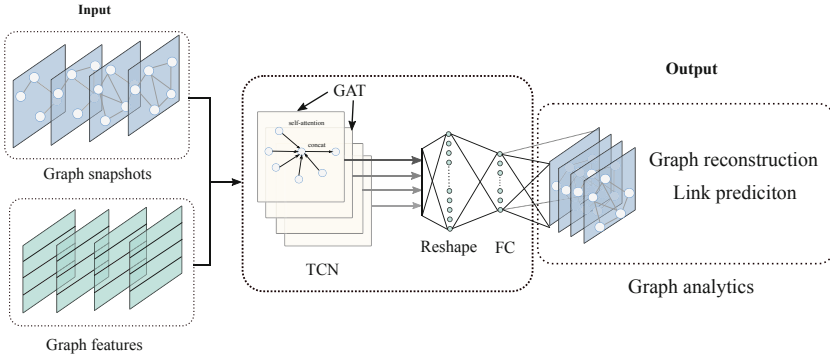
- We present a novel neural architecture named (TemporalGAT) to learn representations on dynamic graphs through integrating GAT, TCN, and a statistical loss function.
- We conduct extensive experiments on real-world dynamic graph datasets and compare with state-of-the-art approaches which validate our method.

## 2 Problem Formulation

In this work, we aim to solve the problem of dynamic graph representation learning. We represent dynamic graph  $G$  as a sequence of graph snapshots,  $G_1, G_2, \dots, G_{\mathcal{T}}$ , from timestamps 1 to  $\mathcal{T}$ . A graph at specific time  $t$  is represented by  $G_t = (V_t, E_t, F_t)$  where  $V_t$ ,  $E_t$  and  $F_t$  represent the nodes, edges and features of the graph respectively. The goal of dynamic graph representation learning is to learn effective latent representations for each node in the graph  $v \in V$  at each time step  $t = 1, 2, \dots, \mathcal{T}$ . The learned node representations should efficiently preserve the graph structure for all node  $v \in V$  at any time step  $t$ .

## 3 TemporalGAT Framework

In this section, we present our proposed TemporalGAT framework, as illustrated in Fig. 1. We propose a novel model architecture to learn representations for dynamic graphs through utilizing GATs and TCNs networks to promote the model ability in capturing temporal evolutionary patterns in a dynamic graph. We employ multi-head graph attentions and TCNs as a special recurrent structure to improve model efficiency. TCNs has proven to be stable and powerful for modeling long-range dependencies as discussed in previous studies [2, 20]. In addition, this architecture can take a sequence of any length and map it to an output sequence of specific length which can be very effective in dynamic graphs due to varying size of adjacency and feature matrices.



**Fig. 1.** The framework of TemporalGAT.

The input graph snapshot is applied to GAT layer which has dilated causal convolutions to ensure no information leakage from future to past graph snapshots. Formally, for an input vector  $x \in \mathbb{R}^n$  and a filter  $f : \{0, \dots, k - 1\} \rightarrow \mathbb{R}$ , the dilated convolution operation  $C_d$  on element  $u$  of the vector  $x$  is defined as:

$$Conv_d(u) = (x *_d f)(u) = \sum_{i=0}^{k-1} f(i) \cdot x_{u-d \cdot i} \tag{1}$$

where  $d$  is the dilation factor,  $k$  is the filter size, and  $u - d \cdot i$  makes up for the direction of the past information. When using a large dilation factors, the output at the highest level can represent a wider range of inputs, thus effectively expanding the receptive field [32] of convolution networks. For instance, through applying dilated convolution operations, it is possible to aggregate the input features from previous snapshots towards final snapshot.

The inputs to a single GAT layer are graph snapshots (adjacency matrix) and graph feature or 1-hot encoded vectors for each node. The output is node representations across time that capture both local structural and temporal properties. The self-attention layer in GAT attends over the immediate neighbors of each node by employing self-attention over the node features. The proposed GAT layer is a variant of GAT [29], with dilated convolutions applied on each graph snapshot:

$$h_u = \sigma \left( \sum_{v \in N_u} \alpha_{vu} W_d x_v \right) \tag{2}$$

where  $h_u$  is the learned hidden representations of node  $u$ ,  $\sigma$  is a non-linear activation function,  $N_u$  represents the immediate neighbors of  $u$ ,  $W_d$  is the shared transformation weight of dilated convolutions,  $x_v$  is the input representation vector of node  $v$ , and  $\alpha_{vu}$  is the coefficient learned by the attention mechanism defined as:

$$\alpha_{vu} = \frac{\exp(\sigma(A_{vu} \cdot a^T [W_d x_v \| W_d x_u]))}{\sum_{w \in N_u} \exp(\sigma(A_{wu} \cdot a^T [W_d x_w \| W_d x_u]))} \tag{3}$$

where  $A_{vu}$  is the edge weight of the adjacency matrix between  $u$  and  $v$ ,  $a^T$  is a weight vector parameter of the attention function implemented as feed-forward layer and  $\parallel$  is the concatenation operator.  $\alpha_{vu}$  is based on softmax function over the neighborhood of each node. This is to indicate the importance of node  $v$  to node  $u$  at the current snapshot. We use residual connections between GAT layers to avoid vanishing gradients and ensure smooth learning of the deep architecture.

Following, we adopt binary cross-entropy loss function to predict the existence of an edge between a pair of nodes using the learned node representations similar to [24]. The binary cross-entropy loss function for certain node  $v$  can be defined as:

$$\mathcal{L}_v = \sum_{t=1}^{\mathcal{T}} \sum_{u \in pos^t} -\log(\sigma(z_u^t \cdot z_v^t)) - W_{neg} \cdot \sum_{g \in neg^t} \log(1 - \sigma(z_v^t \cdot z_g^t)) \quad (4)$$

where  $\mathcal{T}$  is the number of training snapshots,  $pos^t$  is the set of nodes connected with edges to  $v$  at snapshot  $t$ ,  $neg^t$  is the negative sampling distribution for snapshot  $t$ ,  $W_{neg}$  is the negative sampling parameter,  $\sigma$  is the sigmoid function and the dot operator represents the inner product operation between the representations of node pair.

## 4 Experiments

In this section, we conduct extensive experiments to evaluate the performance of our method via link prediction task. We present experiential results of our proposed method against several baselines.

### 4.1 Datasets

We use real-world dynamic graph datasets for analysis and performance evaluation. An outline of the datasets we use in our experiments is given in Table 1.

**Table 1.** Dynamic graph datasets used for performance evaluation.

Dataset	# Nodes	# Edges	# Time steps	Category
Enron	143	2,347	10	Communication
UCI	1,809	16,822	13	
Yelp	6,569	95,361	12	Rating

The detailed dataset descriptions are listed as follows:

- Enron [12] and UCI [21] are online communication network datasets. Enron dataset is constructed by email interactions between employees where the employees represent the nodes and the email communications represent the edges. UCI dataset is an online social network where the messages sent between users represent the edges.

- Yelp<sup>1</sup> is a rating network (Round 11 of the Yelp Dataset Challenge) where the ratings of users and businesses are collected over specific time.

The datasets have multiple graph time steps and were created based on specific interactions in fixed time windows. For more details on the dataset collection and statistics see [24].

## 4.2 Experimental Setup

We evaluate the performance of different baselines by conducting link prediction experiment. We learn dynamic graph representations on snapshots  $S = \{1, 2, \dots, t - 1\}$  and use the links of  $t - 1$  to predict the links at  $t$  graph snapshot. We follow the experiment design by [24] and classify each node pair into linked and non-linked nodes, and use sampling approach to achieve positive and negative node pairs where we randomly sample 25% of each snapshot nodes for training and use the remaining 75% for testing.

## 4.3 Parameter Settings

For our method, we train the model using Adam optimizer and adopt dropout regularization to avoid model over-fitting. We trained the model for a maximum of 300 epochs and the best performing model on the validation set, is chosen for link prediction evaluation. For the datasets, we use a 4 TCN blocks, with each GAT layer comprising attention heads computing 32 features, and we concatenate the output features. The output low-dimensional embedding size of the last fully-connected layer is set to 128.

## 4.4 Baseline Algorithms

We evaluate our method against the several baseline algorithms including static graph representation approaches such as: GAT [29], Node2Vec [7], GraphSAGE [8], graph autoencoders [9], GCN-AE and GAT-AE as autoencoders for link prediction [38]. Dynamic graph representation learning including Know-Evolve [27], DynamicTriad [36], DynGEM [10] and DySAT [24].

## 4.5 Link Prediction

The task of link prediction is to leverage structural and temporal information up to time step  $t$  and predict the existence of an edge between a pair of vertices  $(u, v)$  at time  $t + 1$ .

To evaluate the link prediction performance of each baseline model, we train a logistic regression classifier similar to [36]. We use Hadmard operator to compute element-wise product of feature representation for an edge using the connected

---

<sup>1</sup> <https://www.yelp.com/dataset/challenge>.

**Table 2.** Link prediction results on Enron, UCI and Yelp datasets.

Algorithm	Enron		UCI		Yelp	
	Micro	Macro	Micro	Macro	Micro	Macro
Node2Vec	83.7 ± 0.7	83.1 ± 1.2	80.0 ± 0.4	80.5 ± 0.6	67.9 ± 0.2	65.34 ± 0.2
G-SAGE	82.5 ± 0.6	81.9 ± 0.5	79.2 ± 0.4	82.9 ± 0.2	61.0 ± 0.1	58.56 ± 0.2
G-SAGE + GAT	72.5 ± 0.4	73.3 ± 0.6	74.0 ± 0.4	79.8 ± 0.2	66.2 ± 0.1	65.1 ± 0.2
GCN-AE	81.6 ± 1.5	81.7 ± 1.5	80.5 ± 0.3	83.5 ± 0.5	66.7 ± 0.2	65.8 ± 0.2
GAT-AE	75.7 ± 1.1	76.0 ± 1.4	80.0 ± 0.2	81.9 ± 0.3	65.9 ± 0.1	65.4 ± 0.1
DynamicTriad	80.3 ± 0.8	79.0 ± 0.9	77.6 ± 0.6	80.3 ± 0.5	63.5 ± 0.3	62.7 ± 0.3
Know-Evolve	61.6 ± 1.1	62.3 ± 1.5	71.2 ± 0.5	80.9 ± 0.2	56.9 ± 0.2	59.7 ± 0.2
DynGEM	67.8 ± 0.6	69.7 ± 1.3	77.5 ± 0.3	79.8 ± 0.5	66.0 ± 0.2	66.0 ± 0.2
DySAT	85.7 ± 0.3	86.6 ± 0.2	81.0 ± 0.2	<b>85.8 ± 0.1</b>	70.2 ± 0.1	69.9 ± 0.1
TemporalGAT	<b>86.4 ± 0.4</b>	<b>86.8 ± 0.3</b>	<b>82.7 ± 0.2</b>	85.2 ± 0.2	<b>71.9 ± 0.3</b>	<b>70.3 ± 0.2</b>

pair of nodes as suggested by [7]. We repeat the experiment for 10 times and report the average of Area Under the ROC Curve (AUC) score.

We evaluate each baseline at each time step  $t$  separately, by training the models up to snapshot  $t$  and evaluate the performance at  $t + 1$  for each snapshot up to  $\mathcal{T}$  snapshots. We report the averaged micro and macro AUC scores over all time steps for the methods in Table 2 (given in paper [24]).

From the results, we observe that TemporalGAT outperforms state-of-the-art methods in micro and macro AUC scores. Moreover, the results suggest that GAT using TCN architecture with minimal tuning outperforms graph representation methods, which validates the efficient of TCN in capturing the temporal and structural properties of dynamic graph snapshots.

## 5 Related Work

### 5.1 Static Graph Representation Learning

Static graph embedding can be observed as dimensionality reduction approach that maps each node into a low dimensional vector space which preserves the vertex neighborhood proximities. Earlier research work for linear (e.g., PCA) and non-linear (e.g., IsoMap) dimensionality reduction methods have been studied extensively in the literature [3, 23, 26].

To improve large-scale graph embedding scalability, several approaches have been proposed such as [6, 7, 22], which adopt random walks and skip-gram procedure to learn network representations. Tang et al. [25] designed two loss functions to capture the local and global graph structure.

More recently, network embedding approaches design models that rely on convolutions to achieve good generalizations such as [8, 11, 19, 29]. These methods usually provide performance gains on network analytic tasks such as node classification and link prediction. However, these approaches are unable to efficiently learn representations for dynamic graphs due to evolving nature.

## 5.2 Dynamic Graph Representation Learning

Methods for dynamic graphs representation learning are often an extension of static methods with an additional component to model the temporal variation. For instance, in matrix factorization approaches such as [3, 26] the purpose is to learn node representations that come from eigenvectors of the graph Laplacian matrix decomposition. DANE [16] is based on this idea to update the eigenvectors of graph Laplacian matrix over time series.

For the methods based on random walk such as [7, 22], the aim is to model the node transition probabilities of random walks as the normalized inner products of the corresponding node representations. In [33], the authors learn representations through observing the graph changes and incrementally re-sample a few walks in the successive time step.

Another line of works for dynamic graph representation employ temporal regularization that acts as smoothness factor to enforce embedding stability across time steps [36, 37]. Recent works learn incremental node representations across time steps [10], where the authors apply an autoencoder approach that minimizes the reconstruction loss with a distance metric between connected nodes in the embedding space. However, this may not guarantee the ability of model to capture long-term proximities.

Another category of dynamic graph representation learning is point processes that are continuous in time [13, 17, 28]. These approaches model the edge occurrence as a point process and parameterize the intensity function by applying the learned node representations as an input to a neural network.

More recently, [24] proposed an approach that leverage the most relevant historical contexts through self-attention layers to preserve graph structure and temporal evolution patterns. Unlike this approach, our framework captures the most relevant historical information through applying a temporal self-attention architecture using TCN and GAT layers to learn dynamic representations for real-world data.

## 6 Conclusion

In this paper, we introduce a novel end-to-end dynamic graph representation learning framework named TemporalGAT. Our framework architecture is based on graph attention networks and temporal convolutional network and operates on dynamic graph-structured data through leveraging self-attention layers over time. Our experiments on various real-world dynamic graph datasets show that the proposed framework is superior to existing graph embedding methods as it achieves significant performance gains over several state-of-the-art static and dynamic graph embedding baselines.

There are several challenges for future work. For instance, learning representations for multi-layer dynamic graphs while incorporating structural and feature information is a promising direction.



**Acknowledgments.** This research was supported by Beijing Natural Science Foundation (No. L181010, 4172054), National Key R & D Program of China (No. 2016 YFB0801100), and National Basic Research Program of China (No. 2013CB329605).

## References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations (ICLR) (2015)
2. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint [arXiv:1803.01271](https://arxiv.org/abs/1803.01271) (2018)
3. Belkin, M., Niyogi, P.: Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**(6), 1373–1396 (2003)
4. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: AAAI, pp. 1145–1152 (2016)
5. Chen, J., Ma, T., Xiao, C.: FastGCN: fast learning with graph convolutional networks via importance sampling. arXiv preprint [arXiv:1801.10247](https://arxiv.org/abs/1801.10247) (2018)
6. Fathy, A., Li, K.: ComNE: reinforcing network embedding with community learning. In: Gedeon, T., Wong, K.W., Lee, M. (eds.) ICONIP 2019. CCIS, vol. 1142, pp. 397–405. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-36808-1\\_43](https://doi.org/10.1007/978-3-030-36808-1_43)
7. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864. ACM (2016)
8. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, pp. 1024–1034 (2017)
9. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: methods and applications. arXiv preprint [arXiv:1709.05584](https://arxiv.org/abs/1709.05584) (2017)
10. Kamra, N., Goyal, P., He, X., Liu, Y.: DynGEM: deep embedding method for dynamic graphs. In: IJCAI International Workshop on Representation Learning for Graphs (ReLiG) (2017)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
12. Klimt, B., Yang, Y.: Introducing the Enron corpus. In: CEAS (2004)
13. Kumar, S., Zhang, X., Leskovec, J.: Learning dynamic embeddings from temporal interactions (2018)
14. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data* **1**(1), 2 (2007)
15. Li, J., Dani, H., Hu, X., Tang, J., Chang, Y., Liu, H.: Attributed network embedding for learning in a dynamic environment. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, pp. 387–396. ACM, New York (2017). <https://doi.org/10.1145/3132847.3132919>, <http://doi.acm.org/10.1145/3132847.3132919>
16. Li, J., Dani, H., Xia, H., Tang, J., Liu, H.: Attributed network embedding for learning in a dynamic environment (2017)
17. Nguyen, G.H., Lee, J.B., Rossi, R.A., Ahmed, N.K., Kim, S.: Continuous-time dynamic network embeddings. In: Companion of the The Web Conference 2018, pp. 969–976 (2018)

18. Nguyen, S.H., Yao, Z., Kolbe, T.H.: Spatio-semantic comparison of large 3D city models in CityGML using a graph database (2017)
19. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: International Conference on Machine Learning, pp. 2014–2023 (2016)
20. Oord, A.v.d., et al.: WaveNet: a generative model for raw audio. arXiv preprint [arXiv:1609.03499](https://arxiv.org/abs/1609.03499) (2016)
21. Panzarasa, P., Opsahl, T., Carley, K.M.: Patterns and dynamics of users' behavior and interaction: network analysis of an online community. *J. Am. Soc. Inform. Sci. Technol.* **60**(5), 911–932 (2009)
22. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710. ACM (2014)
23. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500), 2323–2326 (2000)
24. Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H.: Dynamic graph representation learning via self-attention networks (2018)
25. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
26. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000)
27. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 3462–3471 (2017). [JMLR.org](https://jmlr.org)
28. Trivedi, R., Farajtabar, M., Biswal, P., Zha, H.: Representation learning over dynamic graphs (2018)
29. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018). <https://openreview.net/forum?id=rJXMpikCZ>
30. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234. ACM (2016)
31. Yu, A., Dohan, D., Luong, M.T., Zhao, R., Chen, K., Le, Q.: QANet: combining local convolution with global self-attention for reading comprehension (2018)
32. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint [arXiv:1511.07122](https://arxiv.org/abs/1511.07122) (2015)
33. Yu, W., Cheng, W., Aggarwal, C.C., Zhang, K., Chen, H., Wang, W.: NetWalk: a flexible deep embedding approach for anomaly detection in dynamic networks. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2018, pp. 2672–2681. ACM, New York (2018). <https://doi.org/10.1145/3219819.3220024>, <http://doi.acm.org/10.1145/3219819.3220024>
34. Zhang, D., Yin, J., Zhu, X., Zhang, C.: Network representation learning: a survey. arXiv preprint [arXiv:1801.05852](https://arxiv.org/abs/1801.05852) (2017)
35. Zhen, W., Zhang, J., Feng, J., Zheng, C.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
36. Zhou, L., Yang, Y., Ren, X., Wu, F., Zhuang, Y.: Dynamic network embedding by modeling triadic closure process. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

37. Zhu, L., Dong, G., Yin, J., Steeg, G.V., Galstyan, A.: Scalable temporal latent space inference for link prediction in dynamic social networks. *IEEE Trans. Knowl. Data Eng.* **28**(10), 2765–2777 (2016)
38. Zitnik, M., Agrawal, M., Leskovec, J.: Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **34**(13), i457–i466 (2018)