

图神经网络导论

动态图神经网络

周晟

浙江大学 软件学院

2022.12



课程内容

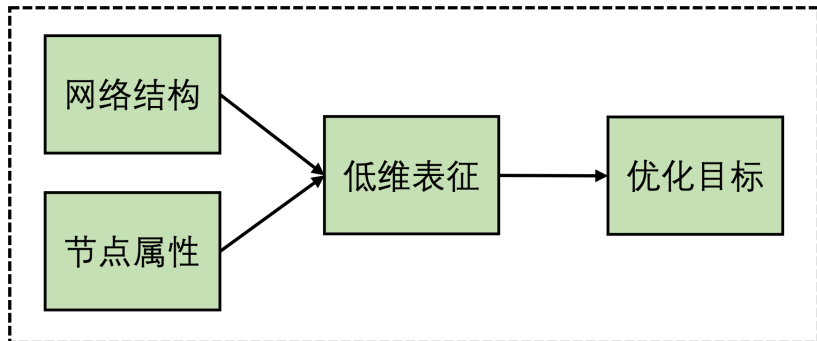
1 研究背景

2 离散动态图神经网络

3 连续动态图神经网络



研究背景



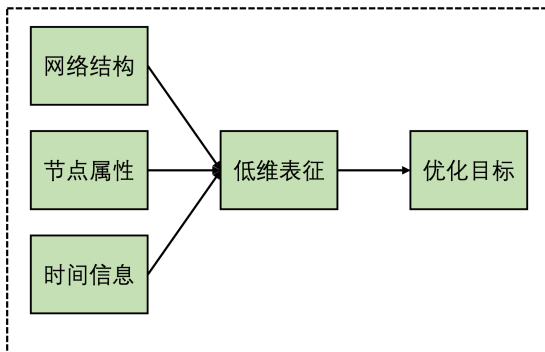
时间信息

真实世界的图中往往带有时间信息



动态图神经网络

动态图神经网络的主要目标是在消息传递机制中融合时间信息，从而捕获图数据的演化特性。



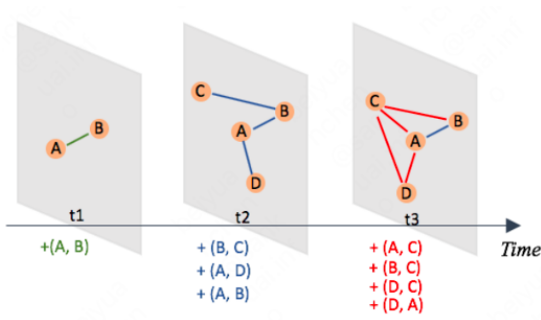
真实世界的图中往往带有时间信息



图上的演化信息

图结构的演化信息

图中的节点、边不是一成不变的，会随着时间变化而动态变化，静态图仅能表示其中某一瞬时状态。



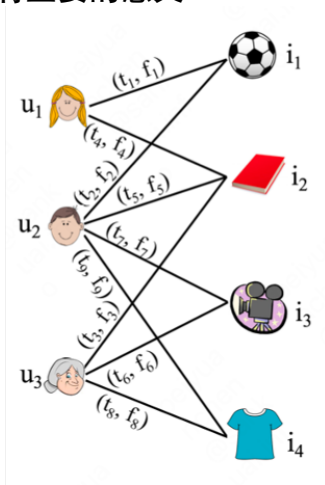
真实世界的图中往往带有时间信息



图上的演化信息

图上的演化信息对图数据挖掘具有重要的意义：

- 1 推荐系统
- 2 异常检测
- 3 社区发现
- 4 疫情控制
- 5 ...



动态图神经网络

动态图神经网络
(Dynamic Graph Neural Networks)

离散型动态图神经网络

连续型动态图神经网络

离散动态图神经网络

将图按时序划分为若干静态图切片 (snapshots) 组成的集合

连续动态图神经网络

通过在节点之间交互的边上体现交互出现的时间和持续时长。

离散/连续动态图神经网络

离散动态图神经网络

- 优点** 可以在整体粗粒度的维度上按照序列的方式进行处理，在单图细粒度的维度上按照静态图结构的方式进行处理。
- 缺点** 对于时序变化的捕获得没有连续型时序图全面，会损失细粒度演化信息。

连续动态图神经网络

- 优点** 对于时序整体变化体现得非常全面，从此类型图中挖掘出更加全面的时序信息和时序演化信号。
- 缺点** 图结构较为庞大复杂，导致在该结构上进行的计算量大，需要更多的同时兼顾算法效率。

1 研究背景

2 离散动态图神经网络

3 连续动态图神经网络



离散动态图神经网络

堆叠型：

对每个离散图snapshot分别作为GNN的输入，将各输出用时间序列模型RNN进行最终聚合。

典型工作：**DGCN**

整合型：

在一个模型层中同时结合GNN和RNN对各个离散图snapshot进行建模。

典型工作：**EvolveGCN**

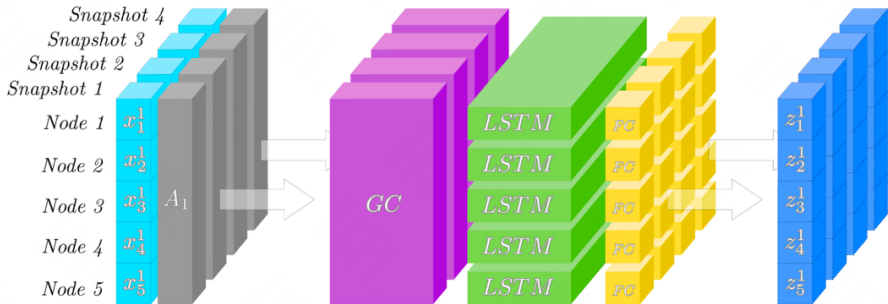
基于深度自编码器：

在一个模型层中同时结合GNN和RNN对各个离散图snapshot进行建模。

典型工作：**DyGEM, dyngraph2vec**



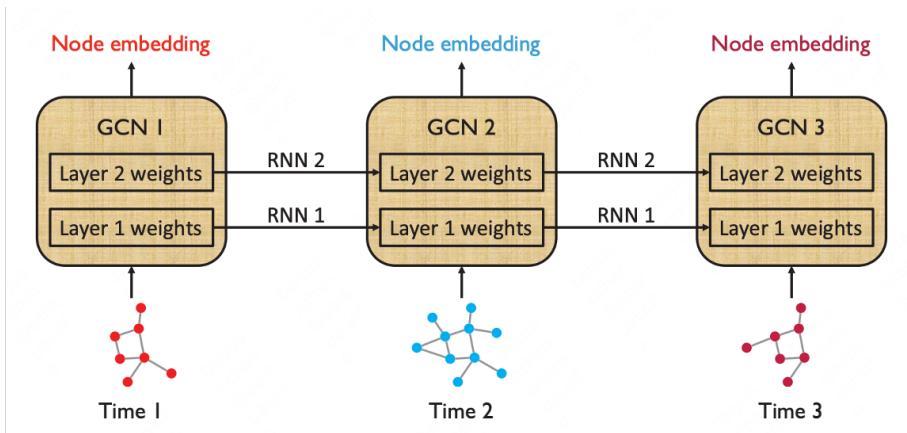
Dynamic Graph Convolutional Networks (PR 2020)



首次将深度神经网络应用于动态图表示中的工作。将 GCN 与 LSTM 相结合，进行离散动态图的建模。

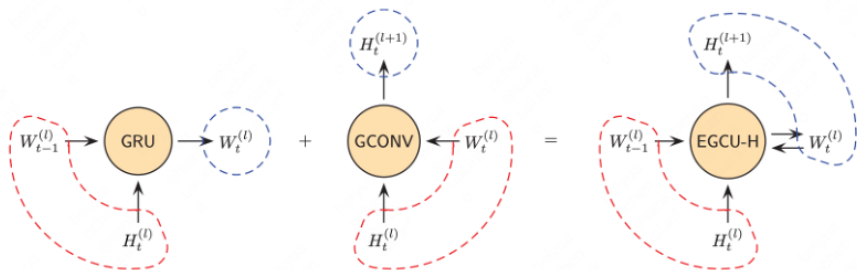


EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs (AAAI 2020)



将 RNN 集成到 GCN 中，每一层 GCN 的权重矩阵（而不是节点的表征）通过 RNN 进行权重演化建模

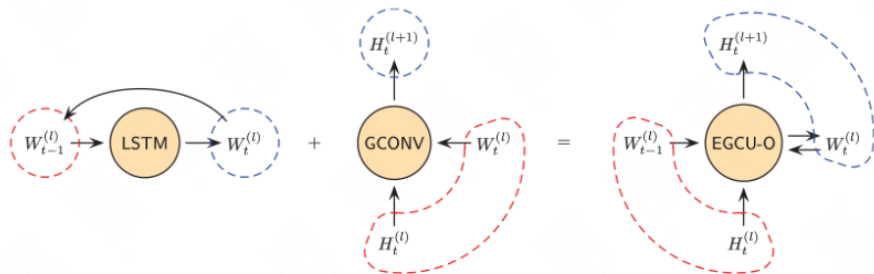
EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs (AAAI 2020)



(a) EvolveGCN-H, where the GCN parameters are hidden states of a recurrent architecture that takes node embeddings as input.

EvolveGCN-H: 将当前层当前时刻的节点表示 H 和当前层上一时刻的权重作为输入，经过 GRU 得到 hidden state 作为新的权重。适用于节点特征信息比较丰富、快速变化的图。

EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs (AAAI 2020)



(b) EvolveGCN-O, where the GCN parameters are input/outputs of a recurrent architecture.

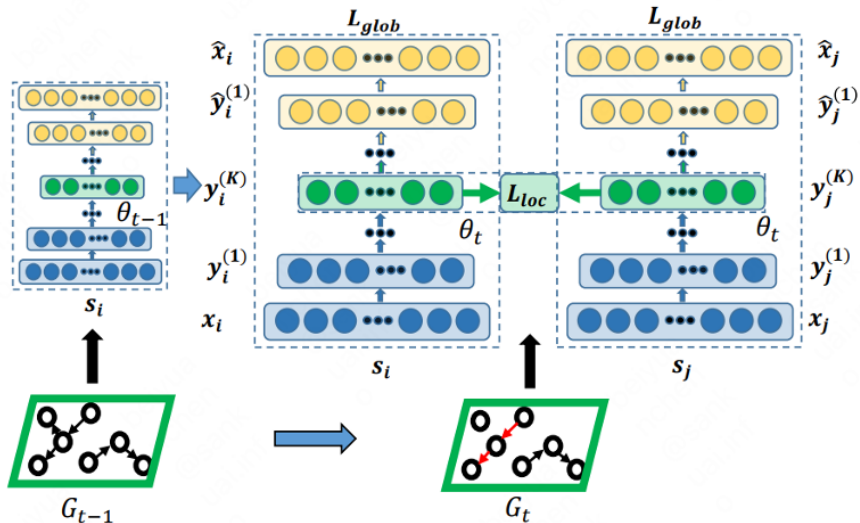
EvolveGCN-O: 只将当前层上一时刻的权重作为输入，经过 LSTM 得到 hidden state 作为新的权重。适用于节点特征信息量不大，而图数据结构起着更重要的作用。

EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs (AAAI 2020)

时间切片 + RNN 的方法存在虽然简单，但是存在如下问题：

- ① 稳定性：因为离散动态图是将一个完整的动态图切分成若干个离散 snapshots，相邻 snapshot 的表征空间可能发生很大变化。
- ② 节点数量增长：动态图随时可能会有新节点加入，现有的方法都是假设各 snapshot 中的节点个数都是一致的，无法处理节点个数变化的情况。
- ③ 规模性：每个 snapshot 单独进行处理的时间复杂度是和切分 snapshot 的粒度和数量有关，难以扩展到很多 snapshots 切片的情况。

DynGEM: Deep Embedding Method for Dynamic Graphs (Arxiv 2018)



1 研究背景

2 离散动态图神经网络

3 连续动态图神经网络



连续动态图神经网络

基于RNN：

在连续型动态图中，通过RNN的方式结合图表示学习获得动态的表示信息。

典型工作：Jodie

基于TPP：

这类模型在一个模型层中同时结合GNN和RNN对各个离散图snapshot进行建模。

典型工作：DyRep

基于时间表示嵌入：

将图中节点的交互时间信息编码到表示向量空间作为模型需学习的一种表示。

典型工作：TGAT, TGN



Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks (KDD 2019)

研究动机

- 1 连续动态图中节点需要经常更新
- 2 连续动态图中需要对节点的未来状态进行预测

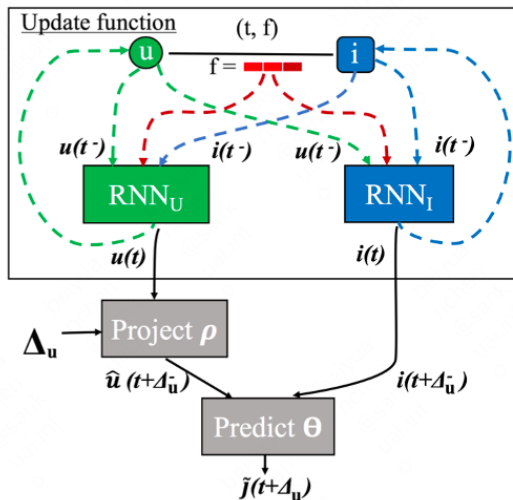
JODIE 通过两个相互作用的 RNN 来更新 embedding:

$$\begin{aligned}u(t) &= RNN_U(u(t^-), i(t^-), \Delta_u, f) \\ i(t) &= RNN_I(i(t^-), u(t^-), \Delta_i, f)\end{aligned}$$

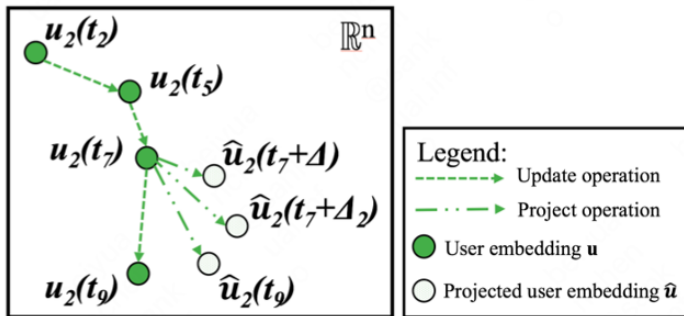
Δ_u 和 Δ_i 分别为 user 和 item 距离上一次交互发生的时间



Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks (KDD 2019)



Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks (KDD 2019)



将时间跨度信息 Δ 映射为向量，并与节点向量融合

$$\hat{u}(t + \Delta u) = (1 + w) * u(t)$$



DyRep: Learning representations over dynamic graphs (ICLR' 19)

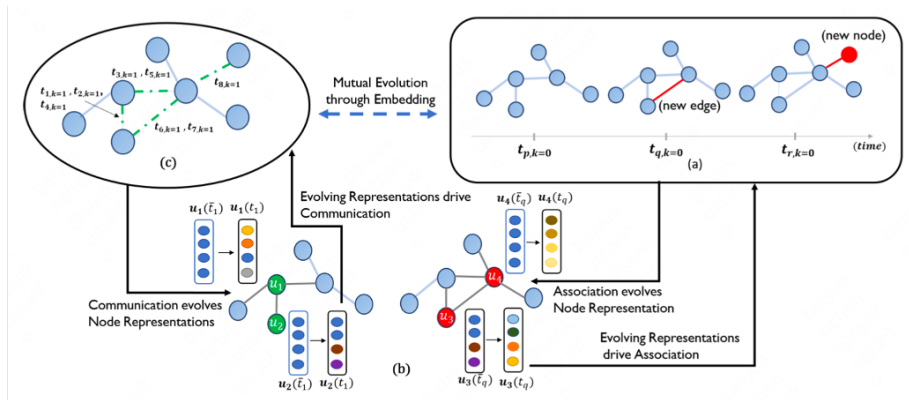
研究动机

真实世界的动态图，动态性体现在两方面：

- ① 拓扑演化（结构层次）：节点和边的数量预计会随着时间的推移而变化，导致图中的结构变化；
- ② 节点演化（节点层次）：与节点之间的活动有关，这些节点可能在结构上连接也可能不连接，使得节点的表示不断变化。



DyRep: Learning representations over dynamic graphs (ICLR' 19)



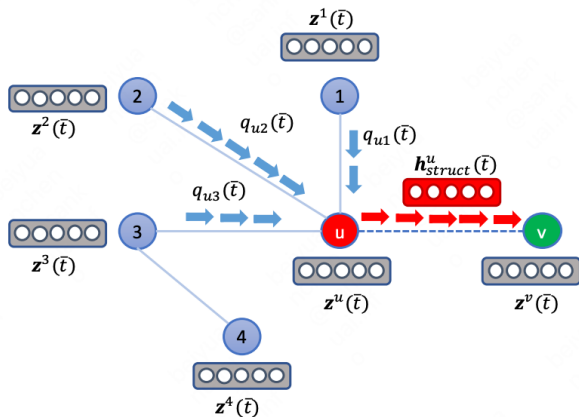
DyRep: Learning representations over dynamic graphs (ICLR' 19)

$$\mathbf{z}^v(t_p) = \sigma(\underbrace{\mathbf{W}^{struct} \mathbf{h}_{struct}^u(\bar{t}_p)}_{\text{Localized Embedding Propagation}} + \underbrace{\mathbf{W}^{rec} \mathbf{z}^v(\bar{t}_p^v)}_{\text{Self-Propagation}} + \underbrace{\mathbf{W}^t(t_p - \bar{t}_p^v)}_{\text{Exogenous Drive}})$$

关键步骤：

- 1 Localized Embedding Propagation (局部嵌入传播)：事件涉及的两个节点形成一条临时或永久路径，使得信息从一个节点的邻居传播到另一个节点
- 2 节点的演变受过去位置的影响，而非随机过程，对各节点自己前一次事件中的 embedding 进行一个线性变换。
- 3 如在涉及节点的两次全局事件之间会有外因的影响，这里体现为时间变化信息，因此进行时间差编码。

DyRep: Learning representations over dynamic graphs (ICLR' 19)



Temporal Point Process Self-Attention:

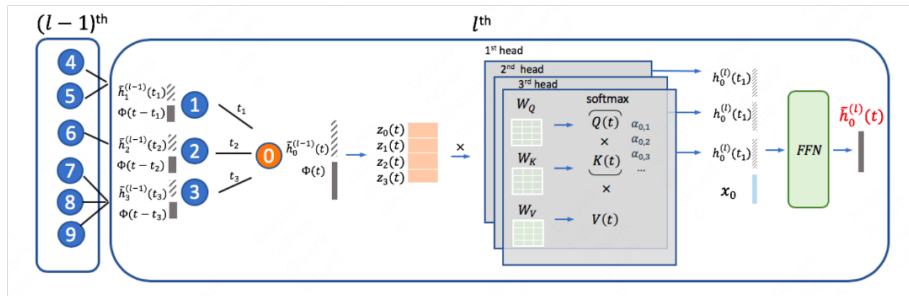
$$\mathbf{h}_{struct}^u(\bar{t}) = \max(\{\sigma(q_{ui}(\bar{t}) * \mathbf{h}^i(\bar{t}))\})$$

$$\mathbf{h}^i(\bar{t}) = \mathbf{W}^h \mathbf{z}^i(\bar{t}) + b^h$$

where $i \in N_u(\bar{t})$ is the node in neighborhood of node u.

$$q_{ui}(\bar{t}) = \frac{\exp(S_{ui}(\bar{t}))}{\sum_{i' \in N_u(\bar{t})} \exp(S_{ui'}(\bar{t}))}$$

Inductive representation learning on temporal graphs (ICLR 2020)



TGAT 的核心模块：时间 Encoding

$$t \mapsto \Phi_d(t) := \sqrt{\frac{1}{d}} [\cos(\omega_1 t), \sin(\omega_1 t), \dots, \cos(\omega_d t), \sin(\omega_d t)]$$

Inductive representation learning on temporal graphs (ICLR 2020)

TGAT 的消息传递机制:

① 邻域信息整合

$$\mathbf{Z}(t) = \left[\tilde{\mathbf{h}}_0^{(l-1)}(t) \parallel \Phi_{d_T}(0), \tilde{\mathbf{h}}_1^{(l-1)}(t_1) \parallel \Phi_{d_T}(t - t_1), \dots, \tilde{\mathbf{h}}_N^{(l-1)}(t_N) \right]$$

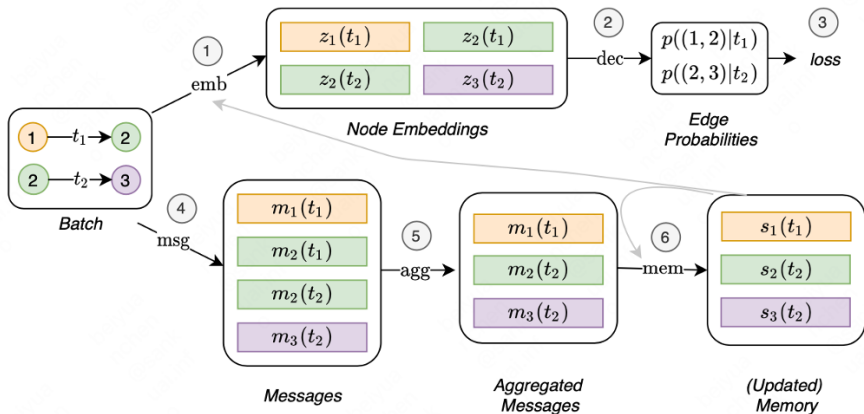
② 时序 Attention

$$\begin{aligned} \mathbf{q}(t) &= [\mathbf{Z}(t)]_0 \mathbf{W}_Q, \mathbf{K}(t) = [\mathbf{Z}(t)]_{1:N} \mathbf{W}_K, \mathbf{V}(t) = [\mathbf{Z}(t)]_{1:N} \mathbf{W}_V \\ \mathbf{h}(t) &= \text{Attn}(\mathbf{q}(t), \mathbf{K}(t), \mathbf{V}(t)) \in \mathbb{R}^{d_h}. \end{aligned}$$

③ 时序消息聚合

$$\begin{aligned} \tilde{\mathbf{h}}_0^{(l)}(t) &= \text{ReLU} \left([\mathbf{h}(t) \parallel \mathbf{x}_0] \mathbf{W}_0^{(l)} + \mathbf{b}_0^{(l)} \right) \mathbf{W}_1^{(l)} + \mathbf{b}_1^{(l)} \\ \mathbf{W}_0^{(l)} &\in \mathbb{R}^{(d_h+d_0) \times d_f}, \mathbf{W}_1^{(l)} \in \mathbb{R}^{d_f \times d}, \mathbf{b}_0^{(l)} \in \mathbb{R}^{d_f}, \mathbf{b}_1^{(l)} \in \mathbb{R}^d \end{aligned}$$

Temporal graph networks for deep learning on dynamic graphs (ICML 2020)



Temporal graph networks for deep learning on dynamic graphs (ICML 2020)

TGN 的核心模块:

① Memory: 保留节点长期的特征, 当新节点出现的时候, 先初始化为全 0 向量, 随着事件的不断出现, 不断更新内容。

② Message Function

① 边事件更新:

$$\mathbf{m}_i(t) = \text{msg}_s(\mathbf{s}_i(t^-), \mathbf{s}_j(t^-), \Delta t, \mathbf{e}_{ij}(t))$$

$$\mathbf{m}_j(t) = \text{msg}_d(\mathbf{s}_j(t^-), \mathbf{s}_i(t^-), \Delta t, \mathbf{e}_{ij}(t))$$

② 节点事件更新:

$$\mathbf{m}_i(t) = \text{msg}_n(\mathbf{s}_i(t^-), t, \mathbf{v}_i(t))$$



Temporal graph networks for deep learning on dynamic graphs (ICML 2020)

TGN 的核心模块:

- 1 Message Aggregator: 对于一个 batch 内出现的节点, 信息汇总为

$$\bar{m}_i(t) = \text{agg}(m_i(t_1), \dots, m_i(t_b))$$

- 2 Memory Updater: 根据当前时刻 t 得到的聚合消息 $\bar{m}_i(t)$, 对相应节点的 memory slot 进行更新:

$$\mathbf{s}_i(t) = \text{mem}(\bar{\mathbf{m}}_i(t), \mathbf{s}_i(t^-))$$

- 3 Embedding:

$$\mathbf{z}_i(t) = \text{emb}(i, t) = \sum_{j \in n_i^k([0, t])} h(\mathbf{s}_i(t), \mathbf{s}_j(t), \mathbf{e}_{ij}, \mathbf{v}_i(t), \mathbf{v}_j(t))$$

Temporal graph networks for deep learning on dynamic graphs (ICML 2020)

	Wikipedia		Reddit		Twitter	
	Transductive	Inductive	Transductive	Inductive	Transductive	Inductive
GAE*	91.44 \pm 0.1	†	93.23 \pm 0.3	†	—	†
VAGE*	91.34 \pm 0.3	†	92.92 \pm 0.2	†	—	†
DeepWalk*	90.71 \pm 0.6	†	83.10 \pm 0.5	†	—	†
Node2Vec*	91.48 \pm 0.3	†	84.58 \pm 0.5	†	—	†
GAT*	94.73 \pm 0.2	91.27 \pm 0.4	97.33 \pm 0.2	95.37 \pm 0.3	67.57 \pm 0.4	62.32 \pm 0.5
GraphSAGE*	93.56 \pm 0.3	91.09 \pm 0.3	97.65 \pm 0.2	96.27 \pm 0.2	65.79 \pm 0.6	60.13 \pm 0.6
CTDNE	92.17 \pm 0.5	†	91.41 \pm 0.3	†	—	†
Jodie	94.62 \pm 0.5	93.11 \pm 0.4	97.11 \pm 0.3	94.36 \pm 1.1	85.20 \pm 2.4	79.83 \pm 2.5
TGAT	95.34 \pm 0.1	93.99 \pm 0.3	98.12 \pm 0.2	96.62 \pm 0.3	70.02 \pm 0.6	66.35 \pm 0.8
DyRep	94.59 \pm 0.2	92.05 \pm 0.3	97.98 \pm 0.1	95.68 \pm 0.2	83.52 \pm 3.0	78.38 \pm 4.0
TGN-attn	98.46 \pm 0.1	97.81 \pm 0.1	98.70 \pm 0.1	97.55 \pm 0.1	94.52 \pm 0.5	91.37 \pm 1.1

DeepSVDD[Ruff et al., 2018]



References I



Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018).

Deep one-class classification.

In *International conference on machine learning*, pages 4393–4402. PMLR.

