Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)

AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN

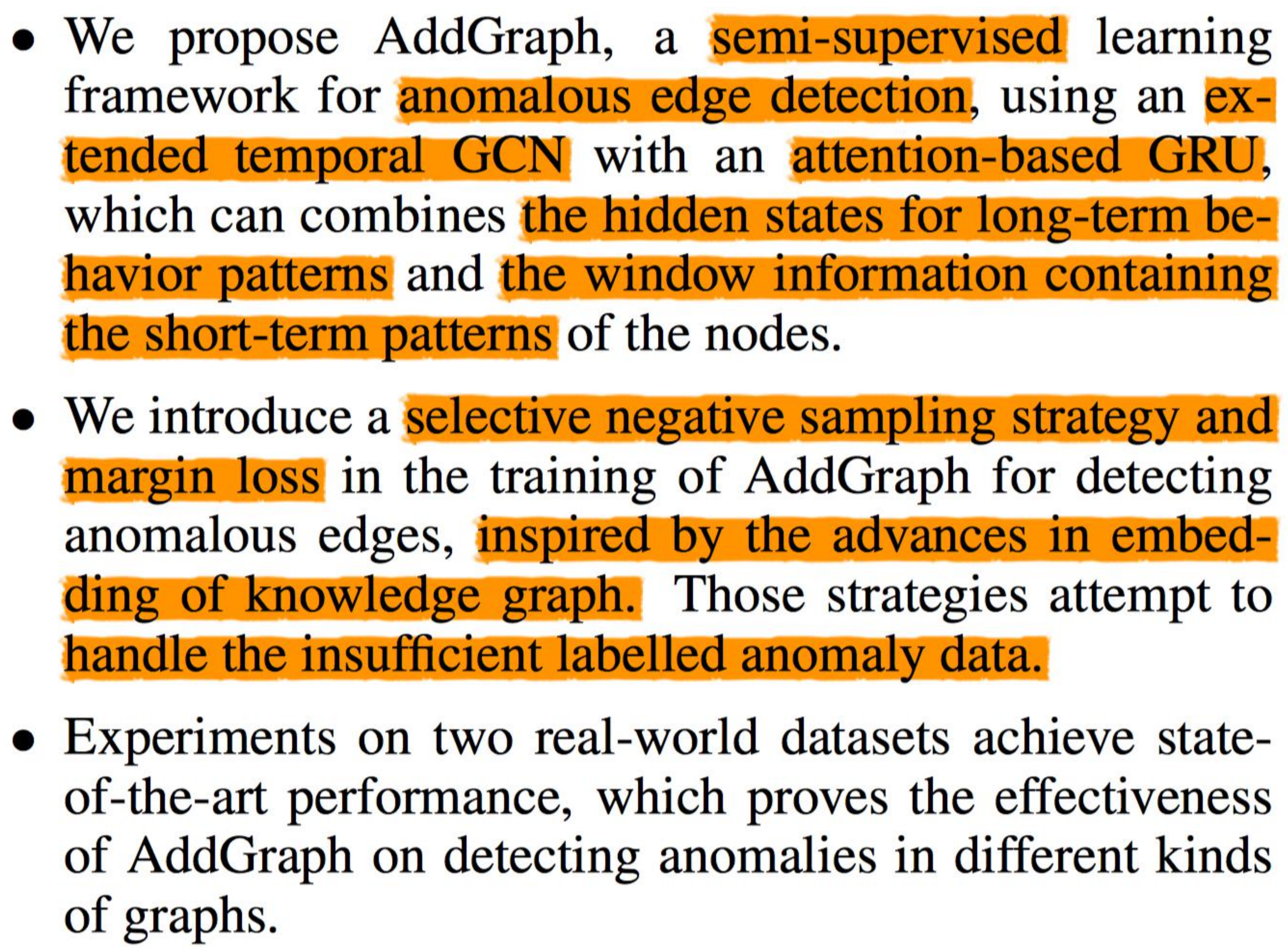
Li Zheng[1,2] , **Zhenpeng Li**[3] , **Jian Li**[3] , **Zhao Li**[3*] and **Jun Gao**[1,2*]
[1]The Key Laboratory of High Confidence Software Technologies, Ministry of Education, China
[2]School of EECS, Peking University, China
[3]Alibaba Group, China
{greezheng, gaojun}@pku.edu.cn, {zhen.lzp,zeshan.lj,lizhao.lz}@alibaba-inc.com

焦乙竹 2019.08.28

- Task: anomalous edges detection (example)
- Data: dynamic graph
- Challenge:
  - flexible and dynamic nature
    - explicit patterns & implicit patterns (dense subgraph)
    - often be normal but hide long-term anomalous behavior (consider structural, temporal and content features)
  - insufficient labelled data

- Shortcomings of traditional methods
  - consider features in a rigid way (deep learning)
  - GCN doesn’ t consider the timing factors
  - dynamic GCN cannot capture the long-term and short-term patterns

- Contributions



- We propose AddGraph, a semi-supervised learning framework for anomalous edge detection, using an extended temporal GCN with an attention-based GRU, which can combines the hidden states for long-term behavior patterns and the window information containing the short-term patterns of the nodes.
- We introduce a selective negative sampling strategy and margin loss in the training of AddGraph for detecting anomalous edges, inspired by the advances in embedding of knowledge graph. Those strategies attempt to handle the insufficient labelled anomaly data.
- Experiments on two real-world datasets achieve state-of-the-art performance, which proves the effectiveness of AddGraph on detecting anomalies in different kinds of graphs.

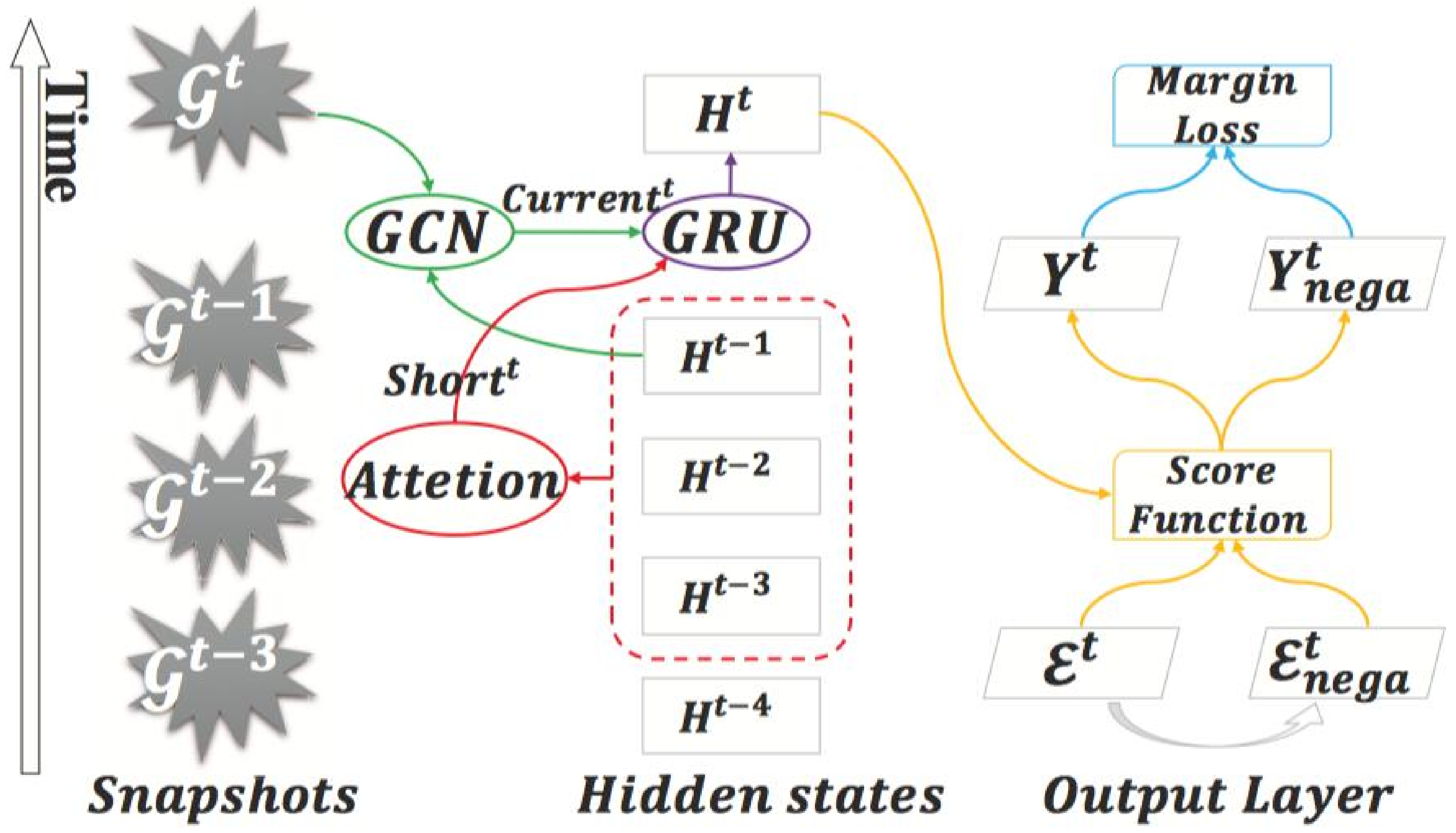


Figure 1: AddGraph framework

$$\mathbf{Current}^t = \mathrm{GCN}_L(\mathbf{H}^{t-1}),$$

$$\begin{aligned}\mathbf{Z}^{(0)} &= \mathbf{H}^{t-1},\\ \mathbf{Z}^{(l)} &= ReLU(\hat{\mathbf{A}}^t\mathbf{Z}^{(l-1)}\mathbf{W}^{(l-1)}),\\ \mathbf{Current}^t &= ReLU(\hat{\mathbf{A}}^t\mathbf{Z}^{(L-1)}\mathbf{W}^{(L-1)}),\end{aligned}$$

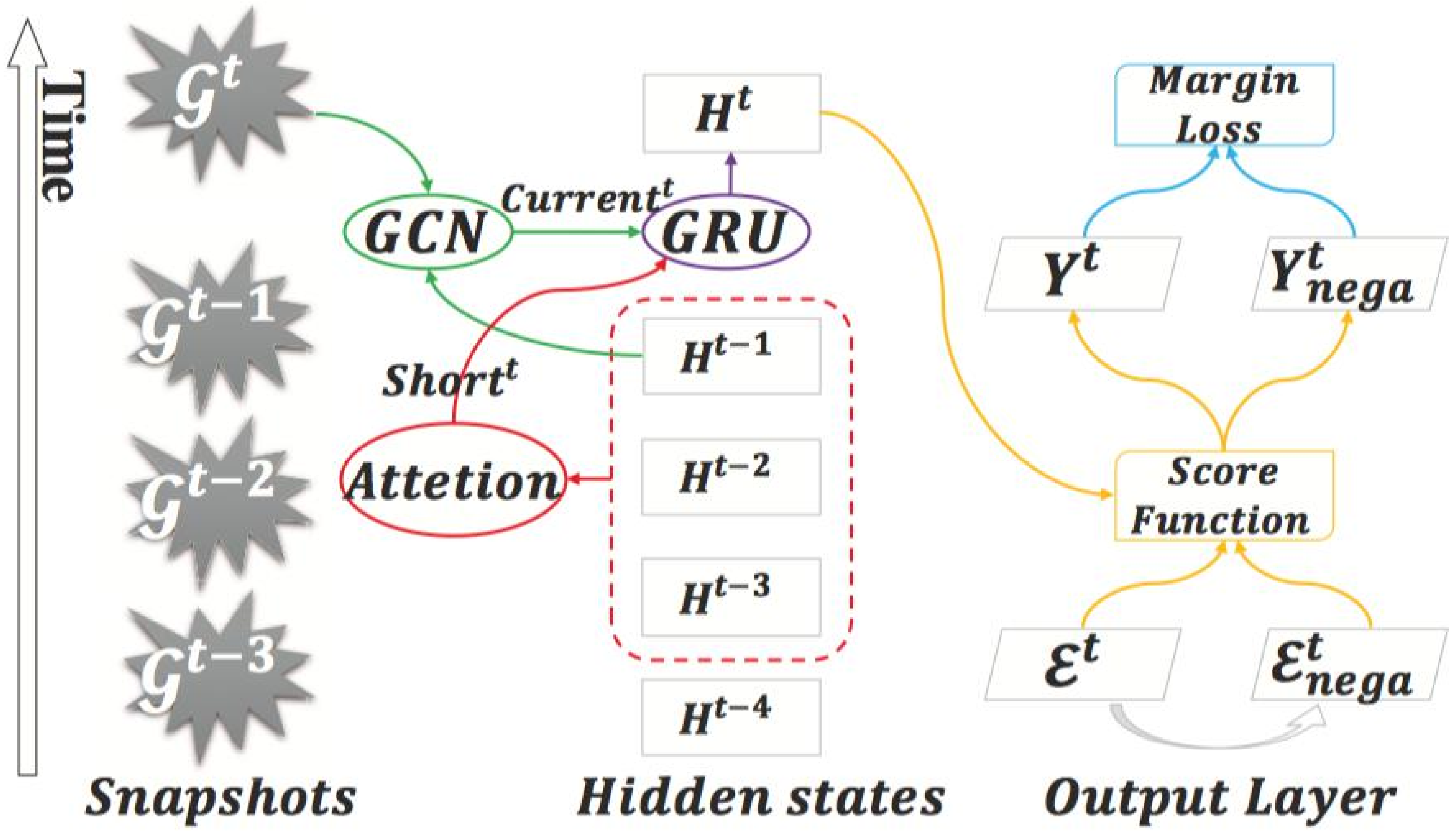


Figure 1: AddGraph framework

$$\mathbf{short}_i^t = \mathrm{CAB}(\mathbf{h}_i^{t-\omega}; \ldots; \mathbf{h}_i^{t-1})$$

$$\mathbf{C}_{h,i}^t = [\mathbf{h}_i^{t-\omega}; \ldots; \mathbf{h}_i^{t-1}] \qquad \mathbf{C}_{h,i}^t \in \mathbb{R}^{\omega \times d}$$

$$\mathbf{e}_{h,i}^t = \mathbf{r}^T \tanh(\mathbf{Q}_h (\mathbf{C}_{h,i}^t)^T) \qquad \mathbf{e}_{h,i}^t \in \mathbb{R}^{\omega}$$

$$\mathbf{a}_{h,i}^t = softmax(\mathbf{e}_{h,i}^t) \qquad \mathbf{a}_{h,i}^t \in \mathbb{R}^{\omega}$$

$$\mathbf{short}_i^t = (\mathbf{a}_{h,i} \mathbf{C}_{h,i}^t)^T \qquad \mathbf{short}_i^t \in \mathbb{R}^{d}$$

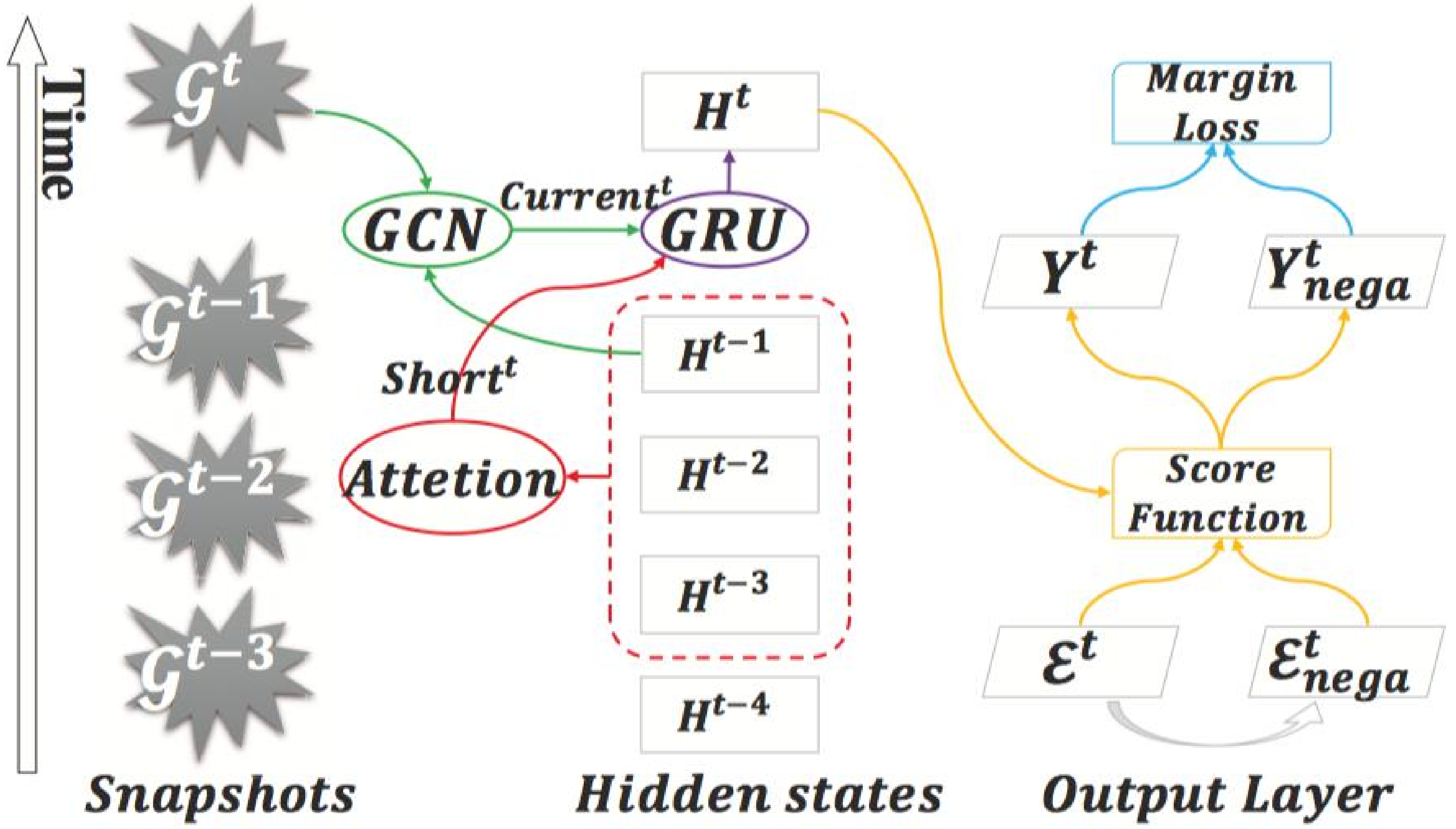


Figure 1: AddGraph framework

$$\mathbf{H}^t = \mathrm{GRU}(\mathbf{Current}^t, \mathbf{Short}^t)$$

$$\mathbf{P}^t = \sigma(\mathbf{U}_P\mathbf{Current}^t + \mathbf{W}_P\mathbf{Short}^t + \mathbf{b}_P)$$
$$\mathbf{R}^t = \sigma(\mathbf{U}_R\mathbf{Current}^t + \mathbf{W}_R\mathbf{Short}^t + \mathbf{b}_R)$$
$$\tilde{\mathbf{H}}^t = \tanh(\mathbf{U}_c\mathbf{Current}^t + \mathbf{W}_c(\mathbf{R}^t \odot \mathbf{Short}^t))$$
$$\mathbf{H}^t = (1 - \mathbf{P}^t) \odot \mathbf{Short}^t + \mathbf{P}^t \odot \tilde{\mathbf{H}}^t$$

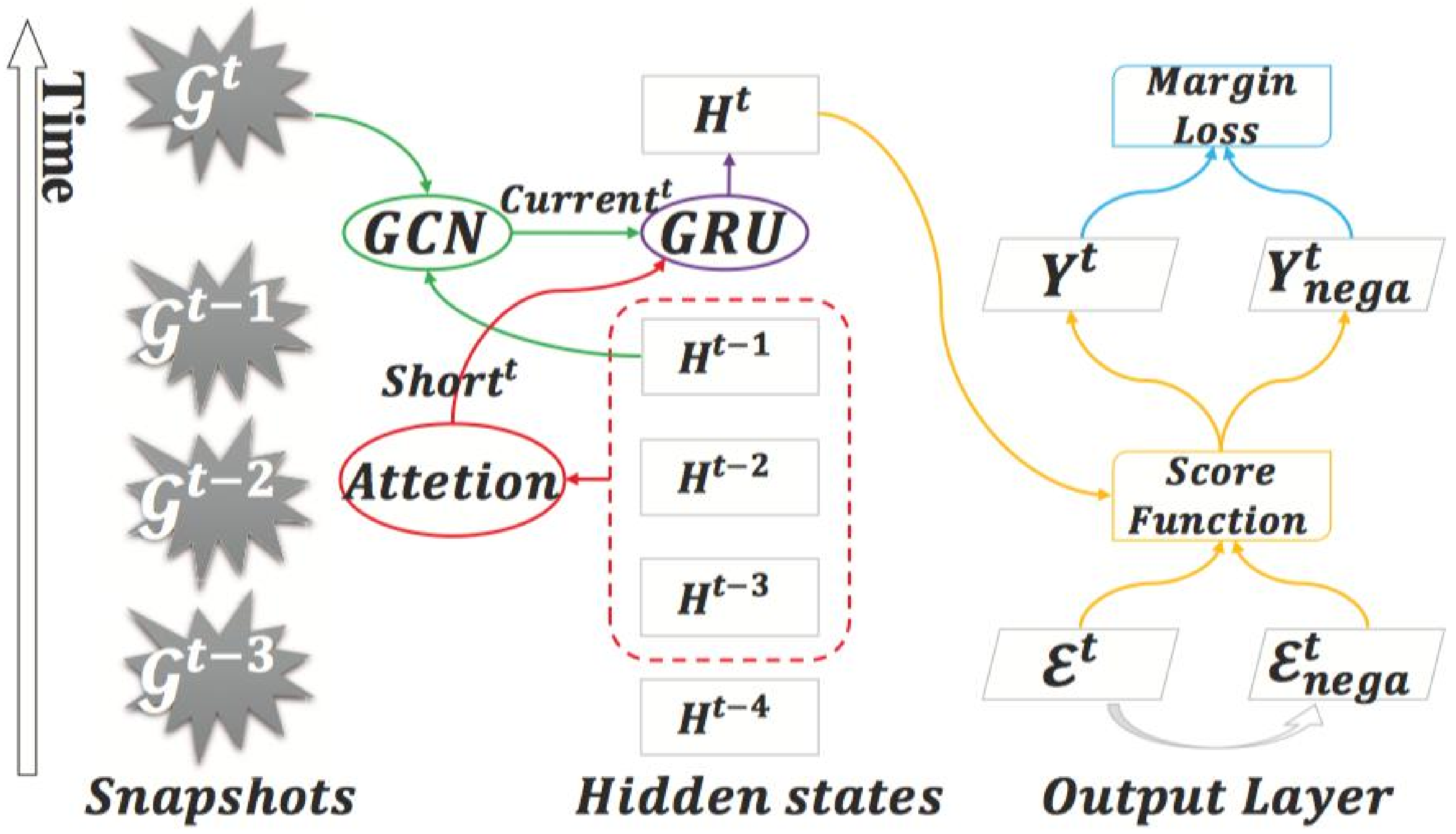


Figure 1: AddGraph framework

Anomalous score computation for edges.

$$f(i, j, w) = w \cdot \sigma(\beta \cdot (||\mathbf{a} \odot \mathbf{h}_i + \mathbf{b} \odot \mathbf{h}_j||_2^2 - \mu))$$

Selective Negative Sampling

- Assumption: all edges in sets at the initial timestamps are normal
- To handle the insufficiency of anomaly data, try to build a model to describe the normal data instead.
- For each normal edge in the graph, we generate a negative sample as an anomalous edge (Bernoulli distribution with parame $\frac{d_i}{d_i+d_j}$)

Margin Loss

- As the generated sampled edges may be still normal, we cannot use a strict loss function such as cross entropy
- $$\mathcal{L}^t = \min \sum_{(i,j,w)\in\mathcal{E}^t} \sum_{(i',j',w)\notin\mathcal{E}^t} \max\{0, \gamma + f(i,j,w) - f(i',j',w)\}$$

$f(i,j,w) > f(i',j',w)$.

- The sampled edge pair is discarded if

Algorithm 1 AddGraph algorithm

```
Input: Edge stream $\{\mathcal{E}^t\}_{t=1}^{T}$
Parameter: $\beta, \mu, \lambda, \gamma, L, \omega, d$
Output: $\{\mathbf{H}^t\}_{t=1}^{T}$
 1: Initialize $\mathbf{H}^0$
 2: repeat
 3:    for $t = 1$ to $T$ do
 4:       Let $\mathcal{L}^t = 0$
 5:       $\mathbf{Current}^t = \mathrm{GCN}(\mathbf{H}^{t-1})$
 6:       $\mathbf{Short}^t = \mathrm{CAB}(\mathbf{H}^{t-w}; ...; \mathbf{H}^{t-1})$
 7:       $\mathbf{H}^t = \mathrm{GRU}(\mathbf{Current}^t, \mathbf{Short}^t)$
 8:       for all $(i, j, w) \in \mathcal{E}^t$ do
 9:          Sample$(i', j', w)$ for $f(i, j, w)$
10:          $\mathcal{L}^t = \mathcal{L}^t + \max(0, \gamma + f(i, j, w) + f(i', j', w))$
11:       end for
12:       $\mathcal{L}^t = \mathcal{L}^t + \mathcal{L}_{reg}$
13:       Minimize $\mathcal{L}^t$
14:    end for
15: until Convergence
16: return $\{\mathbf{H}^t\}_{t=1}^{T}$
```